

# Automatic Identification of Hate Speech on Social Media Platforms using Machine Learning

MSc Research Project  
Cyber Security

Tuvie Akpofure  
Student ID: x18171028

School of Computing  
National College of Ireland

Supervisor: Prof. Christos Grecos

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



<b>Student Name:</b>	Tuvie Akpofure
<b>Student ID:</b>	X18171028
<b>Programme:</b>	Cyber Security
<b>Year:</b>	2019
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Prof. Christos Grecos
<b>Submission Due Date:</b>	12/12/2019
<b>Project Title:</b>	Automatic Identification of Hate Speech on Social Media Platforms using Machine Learning
<b>Word Count:</b>	7135
<b>Page Count:</b>	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

<b>Signature:</b>	
<b>Date:</b>	12/12/2019

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Automatic Identification of Hate Speech on Social Media Platforms using Machine Learning

Tuvie Akpofure  
x18171028

## Abstract

In recent years, the main medium for communication and dissemination of information amongst internet users has been social media platforms such as Facebook, Twitter, etc. These platforms have experienced massive growth and have gained popularity globally based on the features, functions, exposure and benefits attached to them. Furthermore, the increase in popularity of social media platforms has also led to the increase and spread of cybercrimes on these platforms. Some of the cybercrimes that have gained prominence on these platforms are as follow; spamming, phishing, social engineering, etc.

In this research work, our aim is to automatically detect hate speech (which is a type of spam) from tweets sent by users on Twitter. Hate speech could be said to be messages that are offensive, intimidating or insulting targeted at or a group based on their religion, race, ethnicity, sex, etc. Hate speech sometimes leads to physical hate crimes which could be very devastating. Therefore, we used a labelled open source hate speech dataset to help generate features to train our Naïve Bayes model. The model had a precision score of 83% in classifying hate from non-hate comments.

**Keywords:** Hate speech, Machine Learning, Natural Learning Process, Sentiment analysis.

## 1 Introduction

Social media is a sensation that has improved the interaction and communication of internet users worldwide. It mainly refers to software applications and websites that are developed to give internet users the privilege to easily share information quickly, in real-time and with efficiency. The types of social media platforms are as follows; Social networking sites (such as Facebook, Google Plus, etc.), Micro-blogging sites (such as Twitter, Tumblr, etc.), Video Sharing sites (such as YouTube, Vimeo, etc.), Photo sharing sites (such as Instagram, Flickr, etc.) Collaboration tools (Wikipedia, WikiBooks, etc.), Rating or Review sites (such as Amazon ratings), etc. In recent years, social media has had great impact on several areas such as politics, business, dissemination of news, weather reports, user communication, education, health care, etc. Based on reports from Statista [1], at the end of 2013, users of social media platforms were estimated to be 1.61 billion while social media users were predicted to increase to 2.33 billion users globally by the end of 2017. Therefore, due to the global acceptance and usage of social media platforms, cybercrimes such as spamming, phishing, etc. have crept into these platforms making them unsafe for genuine individuals to carry out their normal activities.

Spam is also known as Unsolicited commercial email (UCE), therefore, Spamming is the process of distributing similar inappropriate or unwanted messages across a network to thousands or millions of users without their consent with the aim of gaining the attention of these individuals. Unsolicited bulk email (UBE) can be said to be another form of spam [2]. Spam may be considered dangerous or non-dangerous, it could range from a comic message to the spread of a malicious software which could cause adverse effects to the affected system. Reports generated by Spamhaus [3] and Symantec [4], state that spam is greatly utilized in the distribution of malicious software such as virus, spyware, phishing links, etc. Spamming is

not restricted to only email spamming which is the most popular form of spamming, it involves mobile phone messaging spam, classified ads spam, instant messaging spam (such as WhatsApp and Facebook messenger), internet forum spam, social media spam, network sharing spam, blogging spam, search engine spam, etc.

Spamming has an adverse effect on businesses, social networking platforms, etc. It could lead to loss of productivity and profits for business organizations, it reduces user experience and satisfaction due to incessant disruption, it could lead to legal risks and issues whereby offended recipients file complaints, finally, spam messages could consist of several malware threats which have the ability to steal sensitive information and cause damage to the infected system.

Since social media platforms are part of our daily activities and lives, malicious individuals have seen the need to carry out their malicious activities on such platforms. Cyberattacks which consist of identity theft, social engineering attacks, social spamming, spread of phishing links and websites, etc. are performed by the malicious individuals. In respect to this research work, we will be focusing on social spam and spamming. Social spam can be said to be unsolicited messages sent over a social media platform, web-pages, social bookmarking sites, etc. It consists of the spread of bulk messages, hate speeches, fake news, suspicious links, insults, personal identifiable information, etc. which may be created by individuals or software programs. For spammers to stay hidden and undetected they hide their identities by using fake accounts with fake details to deceive genuine users. Also, they try to update their accounts and spamming techniques to beat systems that try to prevent spamming activities.

Twitter is considered as one of the most popular social media platforms with a large user database. Users have the ability to interact, share news, events, topics and give their opinions on these topics through the twitter interface which allows the typing of just 140 characters. Due to Twitter's popularity, social spamming activities have increased greatly. Spammers take advantage of trending topics to get users to click on URLs that are unrelated to the topics. Also, they use words or hashtags (#) related to trending topics on spam tweets to get the attention of unsuspecting users. The effect social spamming has on social media is great because these spam messages are seen by followers and their corresponding connections. The spam messages that are spread mainly lead to misinformation and misunderstanding. Spammers have several goals which range from profit making to spread of pornography, viruses, political propaganda, phishing or basically just destroy the reputation of the platform. For the purpose of this research work, we will be focusing on Hate speeches on Twitter. According to Merriam Webster dictionary [5], Hate Speech can be said to be a speech or comment aimed at insulting, offending or intimidating a person or a group of people based on certain characteristics such as race, sexual orientation, religion, disability, gender or national origin. This form of spamming is currently plaguing social media platforms because users have the privilege to leave comments and feedbacks which have adverse impacts on the target individual(s). It reduces the overall user experience and it could lead to physical hate crimes. Hence, the need to implement an automatic identification system which distinguishes hate speech texts from non-hate speech texts using Naïve Bayes Machine Learning algorithm.

Machine Learning is been used for this research work because machine learning based applications or systems have the ability to learn, change, grow and improve themselves when they are provided with new data. Also, machine learning algorithms are trained with the data to provide reliable and updated results. Machine learning can be used for Virtual personal assistants, predicting problems, surveillance videos, social media services, filtering of email spam and malware, refining of search engine results, online customer support, online fraud detection, etc. Also, we made use of Natural Language Processing (NLP) which helps in the classification of text data for training and building our model.

This report is aimed at showing if our research question; Can hate speech be automatically detected by machine learning algorithm is answered. Our research objectives are as follows:

- Implement a spammer feature that classifies hate speech comments from non-hate comments.
- Generate text features that can be used to detect hate.
- Train the model with the generated features.

The rest of this report is broken into the following sections; section 2 - focuses on existing approaches used to detect hate speech, section 3 – discusses the general research methodology and the workflow plan for the implementation of this research. Section 4 – this describes the techniques and framework used for the implementation of the model, while, section 5 – the implementation processes were outlined in this section. Section 6 – shows the analysis of the results gotten from our final model then section 7 – concludes and suggests future works.

## **2 Related Work**

Research on online harassment and cyberbullying is still emerging since it's still considered as a new area of study. In recent years, various approaches have been suggested and proposed to help measure and detect offensive or aggressive contents and behaviour on social media platforms. This section focuses on related and existing research works on hate speech activities on social media platforms. It presents methods and approaches that were used to detect and curb this form of spamming activity. It is categorized into two sub-sections which are Non-Machine learning approaches and Machine Learning approaches.

### **2.1 Non-Machine Learning Approaches**

This sub-section focuses on some non-machine learning approaches used to detect hate speech on social media.

Sentence-level Subjectivity Detection can be used to detect hate speeches, where a subjective sentence is said to convey ideas, views, feelings or beliefs. It analyses sentences in a document to check if they are subjective instead of analysing single words. The subjective sentence is then classified to be either positive or negative based on the semantic orientation. Pang and Lee [6] made use of a subjectivity detector model to identify and extract objective sentences from the chosen document. The Min-Cut algorithm used, classified all sentences into subjective or objective. Then minimum cut formulations were used, they also integrated inter sentence level contextual information with the bag of words feature. The performance level of the sentiment classification had a percentage of 86.4.

Ding et al. [7] studied further on the sentiment consistency present within inter-sentential and intra-sentential concepts using natural language expressions. They didn't bother searching for words that were dependent on their domain, instead they proved that a particular word found in the same domain could have different meanings based on the context. Hence, they recommended that aspect and word pairs should be used to derive a context's sentiment. Therefore, the method they used identifies opinion words and their preferences alongside the modified areas.

Ketan S. Modh [8] gives an Indian perspective to controlling hate speech on the internet. He suggests hate speech is an ambiguous concept which depends basically on the cultural and moral ethics of a society and usually narrowed and designed to the interest of the government of a state. Freedom of speech for all citizens is a right that is discussed and protected in the constitution of India but the constitution also gives the state the power to make laws that enforce restrictions on speech if public order, decency or morality, in contempt of a court judgement

are affected. Hence, since the constitution gives the state the power to manage public order, etc., therefore, the state can regulate freedom of speech and there are punishments outlined for people found partaking in hate speech. He stated that according to the Constitution, the government can restrict access to information originating from or available on any computer related device, therefore, organizations or individuals who receive, store or transmit electronic messages for the public are obliged to comply with the government's request to inspect, filter and block digital contents that go through computer related devices. This may seem to be a good approach but this is inefficient in controlling hate speech, because the power the government has could be misused in the sense that it could lead to cyber security issues such as backdoor access by the government which could be exploited by hackers.

## 2.2 Machine Learning Approaches

This sub-section focuses on some machine learning approaches used to detect hate speech on social media platforms. It also focuses on offensive language, since it has similarities to hate speech.

In detecting hate speech, Greevy and Smeaton [9] classify texts with racist contents on web pages by using a supervised machine learning method alongside Bag-of-Words (BoW) feature. Bag-of-Words methodology creates predictive features from words that are within a corpus rather than focusing on word sequences and syntactic/semantic contents. This technique could result in mis-classification which is caused by words that are used in different contexts. Also, using words as primary features for classification has shown that when word sequences are combined into n-grams (word list that occur in sequence from 1-n), the performance of the classifier improves by including degree of context in the features. Nevertheless, an n-gram method may have flaws resulting from high levels of distance between words that are similar or related [10].

In [11], Dadvar et al. classified offensive behavior on YouTube by using profane words in account usernames, made references to profane topics, bullying sensitive subjects, and then used first and second person pronouns. While, in [12], Dinakar et al. worked on detecting offensive contents and hate on social media, they made use of the Bag-of-Words feature and they also included the following as machine learning features; lists of profane words, words with adverse undertones and parts of speech. Also, they went on to include an approach that employs common-sense for classification, this was done by making use of a database that encodes certain pre-existing knowledge about aggressive situations.

Riloff et al. [13] used two bootstrapping algorithms to understand lists of subjective nouns from a collection of unannotated texts. A subjectivity classification model is trained with a small fraction of annotated data which makes use of subjective nouns as features alongside other identified subjectivity characters. They proposed that sentences should be categorized as subjective, if it consists of a subjective expression with an average to high intensity, else, it should be seen as objective. This helps to ascertain that only subjective sentences are classified as subjective. Apart from detecting a sentence's subjectivity and polarity, [14] the strength found in the views and reactions that are conveyed in the clauses could be classified. Syntactic hints and subjectivity characters that have been analyzed in existing research are manipulated to identify the subjectivity strength of a clause.

Burnap et al. [15] designed a rule-based system to classify hostile content on Twitter which is quite similar to the work done in [14] where associational terms were used as machine learning features. In [15], they used terms that were aimed at a target individual or a target group which had to do with accusations and negativity after an adverse social event or occurrence, with the aim of understanding the underlying meaning and context of the terms used. For identification of offensive contents, Chen et al. [10] made use of vulgar language,

obscenities and harsh words as features. These features were weighed depending on the strength of each term as well as how they were used on people. They suggested some rules in modelling offensive contents, this showed an improvement in reduced false negative rates on standard machine learning methodologies.

In [16], Burnap and William stated that “othering” language was a beneficial feature in the classification of hate speech depending on religious beliefs especially for detecting anti-muslim sentiment. Othering could be considered as a recognized form in rhetorical narratives which is focused on hate speech [17], also the ‘we-they’ dichotomy has been recognized to be used in racist conversations. The Bag-of-Words feature was used alongside unigram and bigram features. Some examples of languages that separated certain social groups by geographic regions (example ‘send them home’), made effort to defend the predicted malicious actions from the group (example ‘told you so’), they were openly offensive (example ‘Muslim savages’) on Twitter after the murder of Lee Rigby by Islamist fanatics in London, 2013 [16]. Since identification of othering terms as features for machine classifiers which detect hate against religious groups has been successful, therefore, the aim of their research is to test the efficiency of the ‘us and them’ system on several other forms of hate speech to increase support for the generalization of their proposed method. They also employed the use of the Stanford Lexical Parser with a context free lexical parsing system to help extract othering terms and typed dependencies in a tweet. An F-measure score of 77% was achieved by the three algorithms used which are Random Forest, Support Vector Machine and Bayesian Logistic Regression. Also, in [18], Kwok and Wang focused on detecting hate speech targeted at black people. They made use of Naïve Bayes algorithm and word unigram as their feature for classification. They classified their dataset into three annotators that consisted of people from several racial background to improve the objectivity. The classifier had an average accuracy percentage of 76 on individual tweets. This accuracy score shows that the work done by [18] needs improvement and it could act as a background work for detection of anti-black hate speeches. They also stated that the chance of a tweet being classified as a racist comment is due to the high presence of offensive words. In general, we can say the Bag-of-Words feature is not enough to detect anti-religion hate speeches. Also, in the process of detecting hate speech against race, religion, ethnicity, etc. it is important to make use of bigrams, new vocabularies should be included and a lot of trending or popular hashtags should be used in training the data. Furthermore, the use of supervised approaches in detecting and classifying hate speech tend to jumble it up with offensive language leading to difficulty in the detection of hate speech.

Justin and Tim [19] proposed a Delta Term Frequency–Inverse Document Frequency (TFIDF), which is an intuitive multi-purpose method of weighing word scores effectively before they are classified. For classifying subjectivity, they performed a comparison between the results of Support Vector Machine (SVM) Difference of TFIDFs and Support Vector Machine (SVM) Term Count Baseline. They concluded that the SVM that depends on Delta TFIDF is more accurate and has a low variance. In [20], Robert proved that making use of features that involve readability formulae alongside their combinations, popularly adopted subjectivity clues which could lead to improved accuracy in the classification of sentence-level subjectivity. While, Chenghua et al [21], provided a hierarchical Naïve Bayes model which was dependent on latent Dirichlet allocation, known as subjLDA, for detecting sentence-level subjectivity. The system automatically identified if a particular sentence was expressing thoughts or stating facts.

The study in [16] focused on religion, ethnicity, race, they also made use of the Bag-of-Words feature. The result of their study showed that, Bayesian Logistic regression, Support Vector Machine and Random Forest algorithms had the same F-measure performance score of 77%. Waseem and Hovy [22], chose to compare features that were appropriate for detecting hate speech in English. Word n-gram and Character n-gram were chosen as the main features

for the analysis. The word n-gram constituted of unigram and bigram while, character n-gram recognized each sentence as a bag of character n-gram where all the attributes in the features are seen as strings with “n” as the length. Let’s take an illustration, character 4-grams of “hate speech” will be broken into the following [hate|, |ate\_|, |te\_s|, |e\_sp|, |\_spe|, |spee|, |peec|, and |eech|. Also, [16] made use of additional features such as location and gender. Furthermore, the classification algorithm selected for the study in [22] was the Bayesian Logistic Regression and it gave a result showing that the character n-gram performed better in detecting hate speech than the word n-gram with an accuracy difference of 10%. Pratiwi [23] focused on detecting hate speech aimed at religion in Indonesian language. The selected features are as follows; word unigram and bigram, the amount of hateful words, hateful clauses and the words having negative sentiment were used. The results of Naïve Bayes and SVM were compared, also, a hate-speech dictionary was built by the researcher to help count the amount of words or clauses that had similarities with hate speech. Since the number of tweets that were related to religion were not balanced and did not fall under the non-hate speech category, the dictionary was seen to have a poor result when used as a hate speech dictionary. However, it is more suitable for religion related texts. Alfina et al. [24] decided to detect hate speech in Indonesian language, their aim was to develop a new dataset which comprised of hate speech in general. This dataset contained texts that showed hatred for different religions, race, gender and ethnical groups. They also performed an initial research using machine learning. They stated that machine learning was the most adopted methodology for the classification of texts. Therefore, in the bid to identify hate speech, they evaluated the performance of the various selected features and machine learning algorithms. The extracted features were word n-gram (where n=1 and n=2), character n-gram (where n=3 and n=4) and negative sentiments. The machine learning algorithms that were used for this classification are, Bayesian Logistic Regression, Random Forest Decision Trees, Support Vector Machine (SVM) and Naïve Bayes. 93% F-measure value was achieved when word n-gram feature was used with Random Forest algorithm. Also, the study showed that character n-gram does not perform as good as the word n-gram features.

Samir and Mark [25] aimed at developing learning techniques that build classifiers that can differentiate subjective from objective sentences. They also aimed at formulating methods that are not dependent on linguistic knowledge and can be used for any language. They were able to get the classification of sentence-level subjectivity by employing language independent feature weightings. They used a subjectivity database originating from the reviews of the “Rotten Tomatoes” movie. The supervised machine learning algorithms that were used to detect sentence-level subjectivity are as follows; Fuzzy Control System and Adaptive Neuro-Fuzzy Inference System. Though these machine learning techniques are well known for recognition of patterns, they were used for this work because they haven’t been used for the analysis of subjectivity. They introduced the Pruned ICF Weighting Coefficient which helped in improving the accuracy for detecting subjectivity. The feature extraction process focused on analysing features that were considered informative so as to improve the accuracy of the systems without it having any language related constraint. Since the machine learning models built can be used for any language, therefore, lexical, syntactical analysis and grammatical analysis were not used in this classification. Linguistic knowledge improved the accuracy of the system, hence, this study works for or should be linked with only methods that have the same constraints or make use of the bag of words features and are tested using the same dataset.

In [26] Davidson employed a crowd-sourced hate speech lexicon to gather tweets that contained keywords for hate speech. Crowd-sourcing was used in labelling the tweet sample into three groups; hate speech, offensive language and neither. A multi-classifier model was trained and taught to differentiate or classify the classes. N-grams, TF-IDF and Vader sentiment lexicon were used for feature generation to train the model. They used logistic regression algorithm alongside L1 regularization to decrease the dimensionality of the data, then they

tested several models that were used in already existing works such as, Logistic regression, Naïve Bayes, Decision trees and Linear SVM. The test performed showed that the performance of the Logistic Regression and Linear SVM were considerably better than the other algorithms. The result gotten from their model showed that detailed labelling could help in hate speech detection. It could also pin-point some important challenges faced, therefore, making the classification more accurate. Furthermore, their research showed that racist and homophobic tweets are prone to fall under the hate speech category, sexist tweets could be identified as offensive then tweets that do not have obvious hate keywords are very difficult to classify.

Due to the freedom of expression given to users of social media platforms, the spread of hate, abusive and offensive comments have become popular. Also, these harmful and toxic online texts can result in real time hate crimes. Hence, our need to implement an automatic hate speech detection model. For this paper, we shall make use of the open source dataset used in research [26], also, the approach we shall be using is based on that used in [16] and [26]. We shall categorize the tweets into Hate speech and Non-hate speech and a Naïve Bayes model shall be trained to carry out this classification. We chose Naïve Bayes classification model because it is relatively easier and faster in making predictions when compared to other algorithms, also, it doesn't need a large training dataset to learn features.

### 3 Research Methodology

In this work, we suggest a novel method of automatically detecting hate speech on social media by making use of Natural Language Processing and Machine learning algorithms to examine texts, underlying meaning of tweets and to predict hate. The previous section throws some light on some of the existing works that have been done to detect hate on social media platforms. Furthermore, the goal of this research is to classify tweets into hate and non-hate. Therefore, this section will focus on the process in which the implementation of this research work will be done. The general methodology employed for this work is the Cross-Industry Process for Data Mining methodology approach (CRISP-DM) [27].

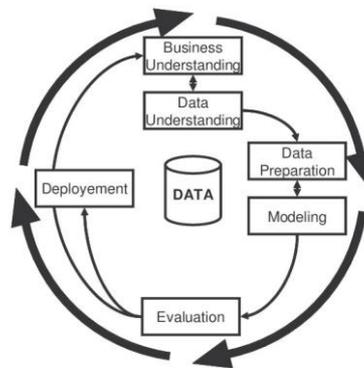


Figure 1: CRISP-DM Methodology<sup>1</sup>

1. In the **Business Understanding** phase, the research objectives and goals are defined and analysed in detail.
2. **Data Understanding** is concerned with collection of the data, analysis of the data and validating the quality of the data.
3. The **Data Preparation** phase involves cleaning of the dataset, constructing, selection of important features and formatting the dataset.
4. The **Modelling** phase focuses on selecting the modelling technique to be used, generating test cases to confirm the quality of the model and building of the model.

<sup>1</sup>[https://www.researchgate.net/publication/258835132\\_A\\_Data\\_Mining\\_Analysis\\_Applied\\_to\\_a\\_Straightening\\_Process\\_Database](https://www.researchgate.net/publication/258835132_A_Data_Mining_Analysis_Applied_to_a_Straightening_Process_Database)

5. **Evaluation** could also be called the testing phase, it involves the implementation of an iterative process which is used to certify the validity of the results obtained from the modelling phase. In this phase, new patterns could be found and new objectives could be set.
6. The results obtained from the aforementioned phases are recorded in the **Deployment** phase, how to manage the results gotten are also concluded and decided in this phase.

The data will be collected from an open source and then examined to validate the features and quality. The Naïve Bayes model will be used as the final classification model. A visual representation of the process flow in which this research will take is below.

### 3.1 Process Flow

The diagram below shows the process and steps taken in the implementation of the objectives of this research. It describes the execution process from the data collection phase to the implementation phase.

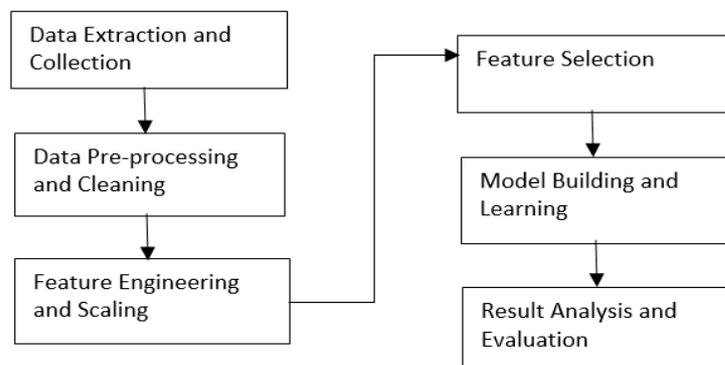


Fig 2: Process Flow

#### Data Extraction and Collection

Data extraction and collection is the first stage towards automatically detecting hate speech, it involves obtaining raw data sets for training and testing. Datasets for hate speech detection usually consist of series of comments or messages of users of the research target platform. Therefore, the dataset used to train our model consists of tweets or comments from Twitter users. It was gotten from Kaggle website which is an open source platform for information. We chose not to collect dataset directly from Twitter because of the platform's new terms of service and General Data Protection Regulation (GDPR).

The dataset used for this research contained 31k tweets but we randomly selected 4000 tweets due to the limitations of the hardware device used. A larger dataset couldn't be used because of the capacity of the physical system which kept running into error. This labelled dataset contained three columns, ID, Tweet and class which had tweets related to religion, race, ethnicity, sexual orientation, national origin and gender. The dataset was annotated automatically to avoid making use of human annotation, like that performed in David et al[16] by CrowdFlower staff.

#### Data Pre-processing and Cleaning

Data pre-processing is a data mining process which focuses on converting raw data into a comprehensible or acceptable format. It prepares raw data for further processing. User

communication over the internet on social media platforms is most times informal, therefore, the appropriate English standards are not met. Raw data is said to be inconsistent, incomplete and noisy, hence, the need to clean and transform it to a format the classification model will understand. User tweets and comments are made up of standard English words, abbreviations, URLs, special characters, slangs, whitespaces, emojis, etc. Therefore, for our data processing and cleaning phase, we created a corpus and then removed stop-words, performed tokenization, removed URL characters, twitter handles and removed special characters such as punctuation marks, etc. Stop-words refer to words that occur the most (such as and, it, is, etc.) but are not necessary or important for the classification process by the model. Furthermore, we converted the tweets to lowercase texts, removed whitespaces and then made use of stemming function to stem our tweets. Stemming helps to reduce the occurrence of words by reducing words to their base words.

### **Feature Extraction and Generation**

Our dataset contained features related to religion, gender, sexual orientation, ethnicity, race and national origin. These features were used for our analysis. We first created a Document Term Matrix (DTM) which designated our document as rows, terms as columns and then showed the frequency in which terms occurred in the tweets. DTM is important because any further analysis to be done on the dataset will be dependent on it. We used the Wordcloud and Barplot visual representation to show the frequent words in the dataset. We decided not to use the Bag of Words (BoW) model, the N-grams and Term Frequency Inverse Document Frequency (TF-IDF) for our feature generation. This decision was based on results from these research works [16], [26] where it was shown that these techniques had difficulties detecting some forms of hate speech such as anti-religion. Rather, for novelty sake, we used Sentiment polarity for our feature generation and sentiment analysis. The sentiment polarity score was used to identify positive and negative tweets. Negative tweets which represent Hate speech were set to “0” while, positive tweets which represent Non-Hate speech were set to “1”. Sentiment analysis could also be referred to as opinion mining or emotion AI. Sentiment analysis is the area of Natural Language Processing (NLP) that helps detect and extract the opinions present in a selected group of texts. It identifies the underlying context of text and classifies it as positive, negative or neutral.

### **Model Building and Learning**

The first step we took in building our model was to decide on the machine learning classification algorithms we were going to train and compare. We made use of the machine learning algorithms; Logistic Regression, Gaussian SVM, Decision Tree and the Naïve Bayes algorithm. For training the model, we split the dataset into training set and test set, where the training set was assigned 80% while the test set 20%. The tweet features generated by the sentiment analysis process were used to train all models on what they should look out for in the classification of hate and non-hate tweets. All packages and libraries needed to run each model were installed and activated. After training the models, we used them to make predictions on the test set. When we compared the results of each model, we observed that the Naïve Bayes classification model had similar results to the other models but it has not really been used for hate speech classification, hence, our reason for making it our final model.

### **Result Analysis and Evaluation**

The result of the implementation showed that our classifier was able to classify words as hate or non-hate easily and quickly. The comparison phase showed that the Naïve Bayes classification model performed similarly to the other models with a precision score of 82% but an F1 score slightly greater than that of other models. From the results derived it is seen that

the accuracy of all models trained is very low, also, the confusion matrix showed that the classifier was quite biased because it tried to classify a lot of tweets as hate. The confusion matrix was also used to show the number of false positives and false negatives the classifier made. We used visualization to represent the test experiments performed on the test set by each model. The visualization was performed on Microsoft Excel.

## 4 Design Specification

This section is used to give a brief description of the architecture used in carrying out the implementation of the proposed hate speech classification model. The several steps that will be performed are shown in the figure below. These steps include the importation of the chosen dataset on to the R environment and the pre-processing and cleaning of the dataset. The next phase will be used to perform feature generation to train the models, it involves the use of sentiment polarity to check polarity scores in the data. In the building of models section, the following models will be built and trained, Logistic Regression, Decision Tree, Naïve Bayes and Gaussian SVM. The results from all models will be compared together and then a visual representation of the results of the final model will be provided.

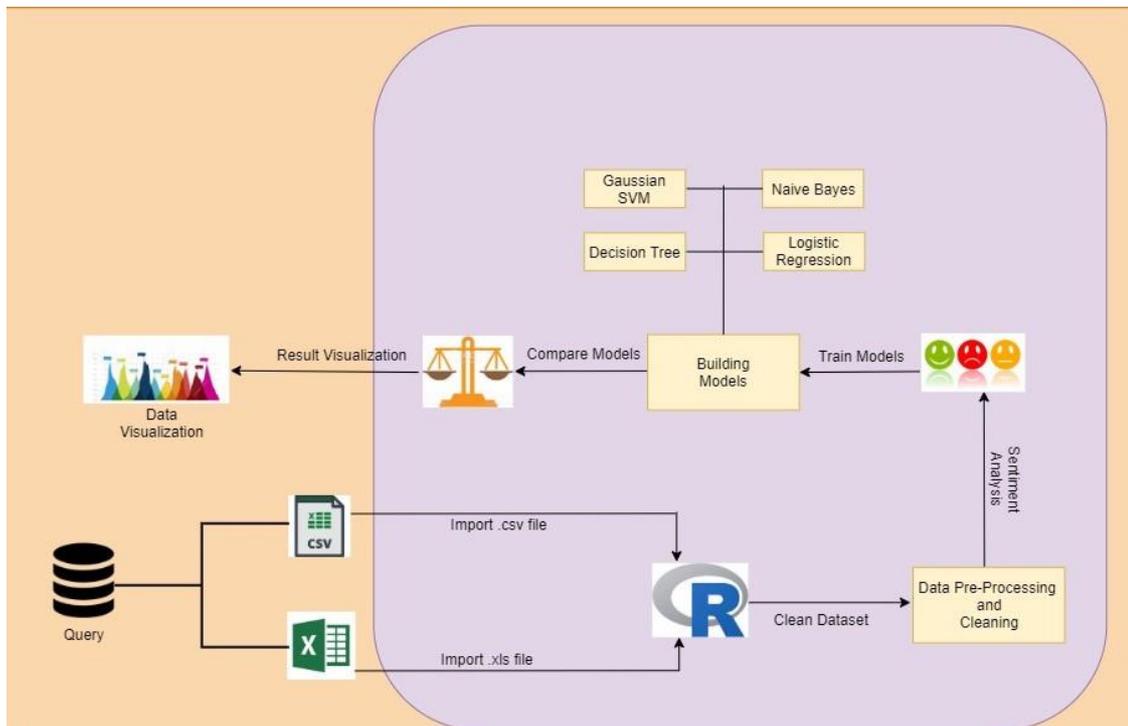


Fig 3: Proposed framework for the Automatic Hate Speech Detection Model

A description of the tools, features and functions used to create the classification model will be discussed at length and in details in the next section.

## 5 Implementation

This implementation section focuses on the setup of the Automatic Hate Speech Detection Model. The final implementation of the model is made up of two files. The files include two .csv dataset files and one .R file. The R file contains the step by step process and implementation of all steps and processes performed towards classifying and building our model were implemented with R programming on RStudio.

For the implementation of this model, we set the working directory on RStudio to the location of the dataset we collected. Since the datasets we used are in .csv formats, we had to import them to the R environment. The techniques and functions used on the R environment all have packages and libraries which help them work effectively. All the necessary packages and libraries were installed and activated before their corresponding functions were called.

The very first step for our implementation is the pre-processing and cleaning phase. This phase helps to remove unwanted and unnecessary features and characters from our dataset, therefore, transforming it to the acceptable format the model understands. A corpus was created for the dataset and the Text mining (tm) package was installed for this pre-processing process. It has tm\_map function that filters the dataset by doing the following;

- removes punctuations,
- removes special characters,
- removes numbers,
- converts the tweets to lowercase texts,
- removes stopwords,
- removes excess whitespaces
- performs stemming on the tweets.

After the pre-processing and cleaning phase, we created a Document Term Matrix (DTM). The DTM was used to convert the earlier created corpus to matrix format, it also assigned columns to each word present in the dataset. The number of time each word occurred in the dataset was represented on this DTM file. Therefore, the use of DTM is important in identifying the frequency in which words occur in a dataset. We used the Wordcloud and barplot functions to show a visual representation of the most occurring words. The wordcloud package was installed and the library activated before the wordcloud function could be used.

The next phase is the feature learning phase, we made use of Sentiment analysis approach to learn and train the dataset on features it should look out for. Sentiment analysis is a branch of Natural Learning Process (NLP), it is the automated method of understanding the context of a subject be it in text format or spoken words. We installed the Sentimentr package and activated the library to help us carry out the sentiment analysis. We created a file with just the Tweet column we needed for the process, then we got the sentiment score by using the get\_sentences function. We used sentiment polarity function to classify the polarity in the tweets. We defined the polarity scores into Negative and Positive tweets, where , if the polarity score of a tweets is  $\leq 0.5$  it is considered “Negative” while, if the polarity score is  $< 0.5$  it is considered “Positive”. We created a polarity table to show the number of tweets that fall into the Negative and Positive categories. The output of the polarity tables showed that 34,636 tweets were Negative and 891 tweets were Positive. We then used “0” to represent Negative tweets which indicate a tweet is Hate speech, while, we used “1” to represent Positive tweets which indicate Non-Hate Speech. We also used a barplot to show a visual representation of the Distribution of Sentiments.

The dataset was split into training and test set, whereby, we used the training set to train our model. The caTools package was installed and the library activated for the split process. 80% of the dataset was assigned to the training set providing the classification model with enough information to learn the features. While, 20% of the dataset was assigned to the Test set to see if the model learned the important features properly. To build our final model, we installed the naïvebayes package, e1071 package, caret package and activated their libraries. We used the traincontrol function to build the model. After the model was built and trained,

we used the predict function to check if the model will properly predict the features on the test set. We went on to use the confusion matrix function to check the number of False positives and False Negatives that occurred on the test set. From the confusion matrix it was shown that about 60% of the tweets were misclassified stating that the model is biased into identifying tweets as hate. Furthermore, we used the following metrics to evaluate the classification model; Accuracy, Recall, Precision and F1 score. The precision score gotten was 82%, the model favored precision over accuracy.

## 6 Evaluation

This section is used to showcase the results gotten from comparing the predictions made by the four machine learning models used. It also focuses on the metrics used for the evaluation of the models. A brief description of the metrics used are as follow;

- Accuracy is the number of correct predictions made divided by the total number of predictions

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

- Precision is the number of True positives divided by the sum of True positives and False positives.

$$\text{Precision} = \frac{tp}{tp + fp}$$

- Recall is the number of True positives divided by the sum of True positive and False negatives.

$$\text{Recall} = \frac{T_p}{T_p + T_n}$$

- F1 score is the average of precision and recall. 0 and 1 are used as F1 scores.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

The table below shows the metric scores for the models. The metric values are compared and it is observed that all models had the same accuracy score of 52%, since accuracy is not enough to check ascertain the performance of the models we used other metrics. The other metrics had very similar values across all models. From the table below, it is observed that the models favored precision over accuracy and the F1 scores are closer to 0 than 1. Since the results gotten from the evaluation metrics are very similar, we chose to you the Time of run for all models to select our preferred model. The Naïve Bayes model had the least runtime of 2 minutes, while, the Gaussian SVM had the longest runtime of 20 minutes.

Models	Accuracy	Precision	Recall	F1
Naïve Bayes	0.52	0.82	0.07	0.13
Logistic Regression	0.52	0.83	0.06	0.12
Decision Tree	0.52	0.83	0.06	0.12
Gaussian SVM	0.52	0.83	0.06	0.12

Table 1: Metrics Table

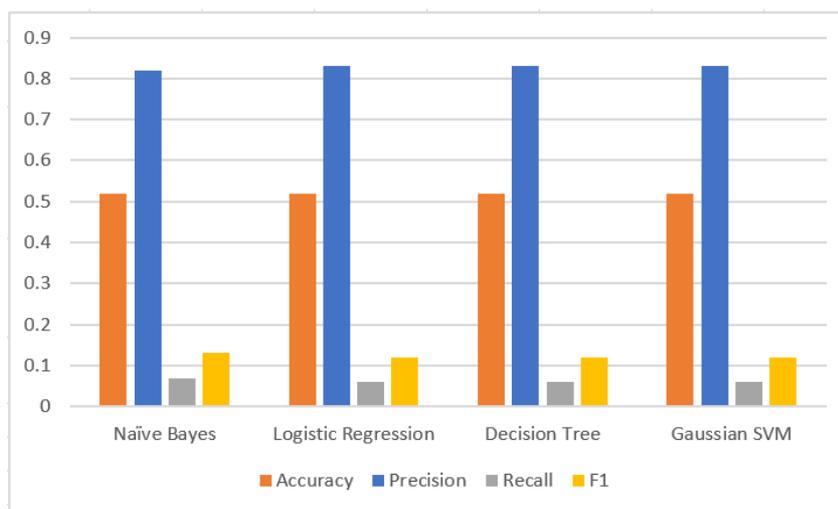


Fig 4: Visual representation of the Metrics

## 6.1 Experiments on the Test Set

The confusion matrix function was used to generate the predictions for the test set. It showed the false positives and false negative values. It was also used to generate the metrics for the models. The trained models were used to predict and classify the text features on the test set using the training gotten from the training set. The graphs below give a visual representation of the predictions done on the test set by all the models. From the graphs, it is observed that all models made very similar predictions on the test set.

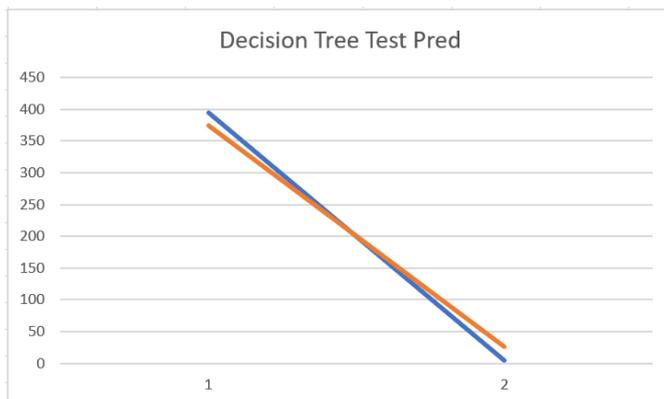


Fig 5: Decision Tree Test Prediction

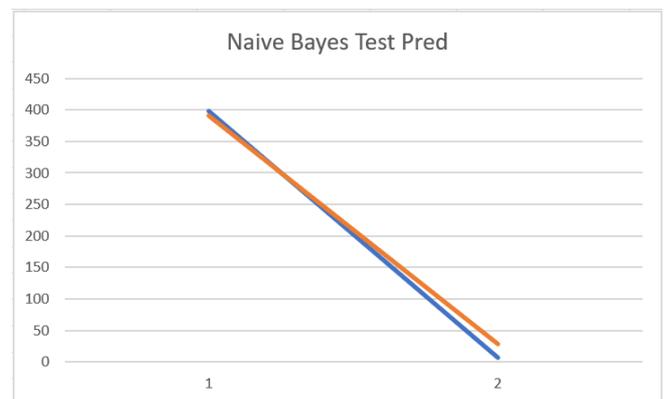


Fig 6: Naïve Bayes Test Prediction

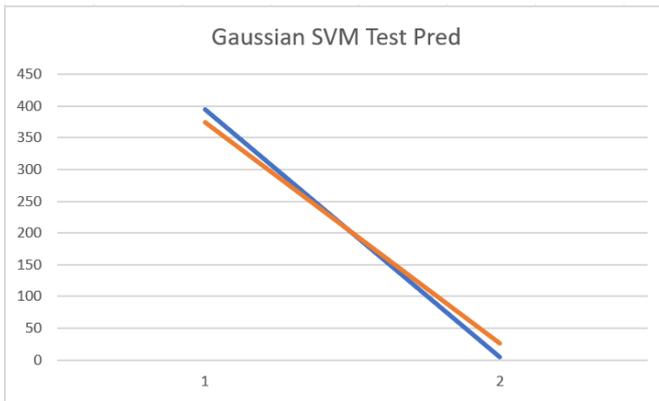


Fig 7: Gaussian SVM Test Prediction



Fig 8: Logistic regression Test Prediction

## 6.2 Discussion

In respect to the results derived from the test predictions by the models, it is obvious from the graphs that the models made a lot of incorrect predictions. A great number of values were predicted by the models to occur but did not occur, while, another great number were not predicted to occur but occurred. Also, from the graph and table, it is shown that the accuracy, precision score, recall score and F1 score are not so high. The average accuracy gotten from all the trained models is 52% which is quite low. Although, accuracy is not enough to ascertain the performance of a model. The precision scores have an average of 83% while the F1 scores have an average of 0.12 for all models. Therefore, due to the close evaluation results, the Naïve Bayes algorithm was chosen as our final model because of its precision and speed at carrying out the classification process.

The design and implementation is good because the model still had the ability to detect hate in the tweets. Therefore, for this work to be improved or reproduced, a dataset that does not contain mainly hate keywords should be used in order to prevent the classifiers from trying to classify comments as hate because of the presence of a hate word. A large dataset with more texts could be considered, also, the hardware device to be used should run on a fast processor and have large memory because text analysis takes up a lot of system resources. Other improved classification models (such as XGBoost) could be used to solve this problem in order to compare the results with the ones already used in this research. Then for feature generation, other approaches such as N-grams should be employed.

## 7 Conclusion and Future Work

In conclusion, we were able to build a classification model which differentiates hate speech from non-hate speech. Also, we generated text features which were used to train our models for the classification. The final classification model built answered our request question which says, can machine learning be used to detect hate speech on social media platforms? In respect to existing works, it is observed that certain words are relevant when classifying hate and non-hate speech. Tweets that were tagged hate had demeaning racist, sexist and homophobic words. Though this allows us to easily detect hate and offensive comments, it may also cause our model to misclassify terms if they are lacking these well-known hate terms. Also, from our findings, it can be said that hate speech can be directed at an individual, a group or it could be used without any direct target. The models used all had average accuracy and fairly high

precision scores, therefore, the performance of the final model was quite low. Also, the hardware device used for this work gave a lot of issues due to its physical properties.

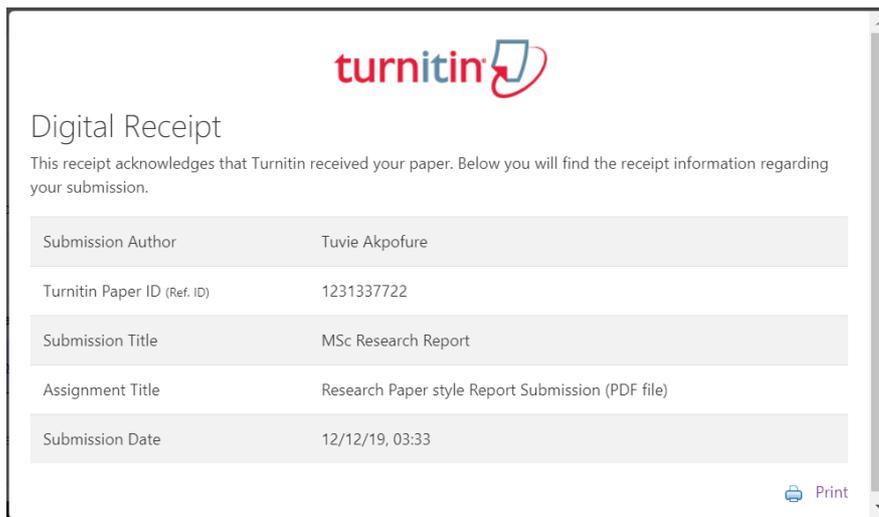
For future work, researchers should make use of datasets that have hateful texts and contexts rather than using datasets that are basically consist of the popular hate keywords. The works should be able to build models with better accuracy and performance. Also, they should examine the attributes and motivation of users that involve in spreading hate on social platforms.

## References

- [1] Statista, "Statista," Statista, [Online]. Available: <http://www.statista.com/>. [Accessed September 2019].
- [2] Cranor, L. F. & LaMacchia, B. A., "Spam!," *Communications of the ACM*, vol. 41, no. 8, pp. 74-83, 1998.
- [3] Spamhaus, "The definition of spam," Spamhaus, July 2006. [Online]. Available: <http://www.spamhaus.org/definition.html>. [Accessed October 2019].
- [4] Symantec, "Symantec internet security threat report," Symantec, March 2006. [Online]. Available: <http://www.symantec.com/enterprise/threatreport/index.jsp>. [Accessed October 2019].
- [5] Merriam Webster Dictionary, "Merriam Webster Dictionary," Merriam Webster Dictionary, [Online]. Available: <https://www.merriam-webster.com/dictionary/hate%20speech>. [Accessed October 2019].
- [6] Pang, and Lee, L. , "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," in *Association of Computational Linguistics (ACL)* , 2004.
- [7] Ding, X., Liu, B. and Yu P. , "A holistic lexicon-based approach to opinion mining," in *Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*, 2008.
- [8] Modh, Ketan, "Controlling Hate Speech on the Internet: The Indian Perspective," Available at SSRN, 12 October 2015. [Online]. Available: <https://ssrn.com/abstract=2783447>. [Accessed October 2019].
- [9] Greevy, E., and Smeaton, A.F., "Classifying Racist Texts Using a Support Vector Machine," in *In Proceedings of the 27th Annual International Conference on Research and Development in Information*, Sheffield, UK, 2004.
- [10] Chen, Y., Zhou, Y., Zhu, S. and Xu, H. , "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety," in *In Proceedings of the Fourth ASE/IEEE International Conference on Social Computing (Social-Com 2012)*, September 3-6, , Amsterdam., 2012.
- [11] Dadvar, M., Trieschnigg, D. and De Jong, F. , "Expert Knowledge for Automatic Detection of Bullies in Social Networks," in *In Proceedings of the 25th Benelux Conference on Artificial Intelligence, BNAIC* , 57-64., Delft, Netherlands, 2013, November 7-8.
- [12] Dinakar, K., Jones, B., Havasi, C., Lieberman, H. and Picard, R. , "Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying," *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2 (3): Article 18., 2012.
- [13] Riloff, E. and Wiebe, J., "Learning extraction patterns for subjective expressions," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)* . , 2003.
- [14] Wilson, T., Wiebe, J. and Hwa, R. , "Just how mad are you? Finding strong and weak opinion clauses," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)* . , 2004.
- [15] Burnap, P., Rana, O., Avis, N., Williams, M.L., Housley, W., Edwards, A., Morgan, J. and Sloan, L., "Detecting Tension in Online Communities With Computational Twitter Analysis" *Technological Forecasting and Social Change*, 2013.
- [16] Burnap P, Williams ML, "Cyber hate speech on Twitter: an application of machine classification and statistical modelling for policy and decision making. *Policy & Internet* 7(2)," pp. 223-242, 2015.
- [17] Meddaugh PM, Kay J, " Hate speech or 'reasonable racism?' The other in stormfront," in *J Mass Media Ethics* 24(4):251-268, 2009.
- [18] I. Kwok and Y. Wang, "Locate the Hate: Detecting Tweets against Blacks," in *Twenty-Seventh AAAI Conf. Artif. Intell.*, pp. 1621-1622, 2013., 2013.
- [19] a. T. F. Justin Martineau, "Delta TFIDF: An Improved Feature Space for Sentiment Analysis," in *In Proceedings of the 3rd AAAI International Conference on Weblogs and Social Media.*, 2009.
- [20] Robert Remus, "Improving Sentence-level Subjectivity Classification through Readability Measurement," in *NODALIDA-2011 Conference Proceedings*, pp. 168-174., 2011.

- [21] Chenghua Lin, Yulan He and Richard Everson, "Sentence Subjectivity Detection with Weakly-Supervised Learning," in *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pp. 1153–1161, 2011.
- [22] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in *Proc. NAACL Student Res. Work*, pp. 88–93, 2016.
- [23] S. H. Pratiwi, "Detection of Hate Speech against Religion on Tweet in the Indonesian Language Using Naive Bayes Algorithm and Support Vector Machine," B.Sc. Tesis, Universitas Indonesia, , Indonesia, 2016.
- [24] S. Rustamov and M. A. Clements , "Computational Approaches to Subjectivity, Sentiment and Social Media Analysis,," *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Atlanta, Georgia*, p. 108–114, 14 June 2013.
- [25] Davidson, T., Warmesley, D., Macy, M., Weber, I. , "Automated Hate Speech Detection and the Problem of Offensive Language," in *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*,, Montreal, Canada, 2017.
- [26] Alfina, I., Mulia, R., Fanany, M. I., and Ekanata, Y. , "Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE., 2017.
- [27] Caprace, J., Losseau, N., Archambeau, D., Bair, F & Philippe, R., "A Data Mining Analysis Applied to a Straightening Process Database," pp. 415-425., 2007.

## Submission Receipt



The image shows a digital receipt from Turnitin. At the top is the Turnitin logo. Below it, the text reads "Digital Receipt" and "This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission." The receipt information is presented in a table with five rows, each with a label and a value. At the bottom right of the receipt area is a "Print" button with a printer icon.

Submission Author	Tuvie Akpofure
Turnitin Paper ID (Ref. ID)	1231337722
Submission Title	MSc Research Report
Assignment Title	Research Paper style Report Submission (PDF file)
Submission Date	12/12/19, 03:33

Print