Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A probabilistic clustering model for hate speech classification in twitter

Femi Emmanuel Ayo [a,*], Olusegun Folorunso [b], Friday Thomas Ibharalu [b], Idowu Ademola Osinuga [c], Adebayo Abayomi-Alli [b]

[a] *Department of Physical and Computer Sciences, McPherson University, Seriki Sotayo, Ogun State, Nigeria*
[b] *Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria*
[c] *Department of Mathematics, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria*

ARTICLE INFO

Keywords:
Twitter
Hate speech
Fuzzy logic
Combinatorial algorithm
Bayesian function
Sentiment analysis

ABSTRACT

The key challenges for automatic hate-speech classification in Twitter are the lack of generic architecture, imprecision, threshold settings and fragmentation issues. Most studies used binary classifiers for hate speech classification, but these classifiers cannot really capture other emotions that may overlap between positive or negative class. Hence, a probabilistic clustering model for hate speech classification in twitter was developed to tackle problems with hate speech classification. A metadata extractor was used to collect tweets containing hate speech keywords and a crowd-sourced experts was employed to label the collected hate tweets into two categories: hate speech and non-hate speech. Features representation was done with Term Frequency- Inverse Document Frequency (TF-IDF) model and enhanced with topics inferred by a Bayes classifier. A rule-based clustering method was used to automatically classify real-time tweets into the correct topic clusters. Fuzzy logic was then used for hate speech classification using semantic fuzzy rules and a score computation module. From the evaluation results, it was observed that the developed model performed better in hate speech detection with F1-sore of 0.9256 using a 5-fold cross validation. Similarly, the developed model for hate speech classification performed better with F1-score of 91.5 compared to related models. The developed model also indicates a more perfect test having an AUC of 0.9645, when compared to similar methods. The Paired Sample t-Test validated the efficiency of the developed model for hate speech classification.

1. Introduction

Microblogging is a phenomenon in social media that enables big data to be generated by means of short digital contents. Twitter is an online service that enables users to share electronic short text (tweets) limited to 140 characters (Atefeh & Khreich, 2015; Daniel et al., 2017; Schnitzler et al., 2016). Twitter's messaging services like portability, instant messaging, and ease of use has made scholars in various fields to move their research focus to Twitter (Atefeh & Khreich, 2015; Earl & Garrett, 2016; Medina & Diaz, 2016; Lu, 2018). A one-sided relationship between users and followers known as asymmetric relationship is allowed on Twitter where a user can connect to other users without their acknowledgment (Brzozowski & Romero, 2011; Taberner, 2016; Bonini et al., 2016; Sankaranarayanan, Samet, Teitler, Lieberman, & Sperling, 2009). Research suggests that together with the excellent messaging systems, this asymmetric relationship accounts for the success and increased interest in Twitter. Moreso, some scholars believe this asymmetric relationship has made users more accessible to celebrities. Twitter's increased accessibility has increased its use and ranked it as the second most popular social network with over 3.7 million active users tweeting around 10 million tweets per day (Nejad et al., 2020).

Twitter allows for a series of tweets to form a complete story because of its message length constraint. This series of tweets therefore mainly includes both unwanted and desirable material emanating mainly from abbreviations of terms and embedded references to other stories (Hurlock & Wilson, 2011; Kwak et al., 2010). Twitter's openness in allowing users to express their feelings and emotions about a debatable occurrence has contributed to big data being generated and unnecessary content also increasing. Companies are increasingly using Twitter to promote and recommend goods, brands and services, build and maintain reputations, evaluate the feelings of consumers about their products, respond to complaints from customers, and enhance decision-making

[*] Corresponding author.
E-mail addresses: ayofe@mcu.edu.ng (F.E. Ayo), folorunsoo@funaab.edu.ng (O. Folorunso), ibharaluft@funaab.edu.ng (F.T. Ibharalu), osinuga08@gmail.com (I.A. Osinuga), abayomiallia@funaab.edu.ng (A. Abayomi-Alli).

https://doi.org/10.1016/j.eswa.2021.114762
Received 14 February 2020; Received in revised form 12 February 2021; Accepted 18 February 2021
Available online 24 February 2021
0957-4174/

and market intelligence (Jansen et al., 2009; Liu et al., 2012; Pak & Paroubek, 2010). Twitter has also emerged as a strong channel of communication to gather and disseminate news (Lee, Lee & Choi, 2020), to forecast outcome of elections and to exchange political events and discussions (Grover, Kar, Dwivedi & Janssen, 2019). It has also become an important analytical tool for crime forecasting, tracking terrorist activities and detecting hate speech (Wang et al., 2012).

Hate speech is commonly defined as any message that mocks or discriminates against a person or group based on specific characteristics such as colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics (Nockleby, 2000). The amount of hate speech is steadily increasing due to Twitter's popularity and the resulting big data from user-generated content (Gitari, Zuping, Damien & Long, 2015).

Nevertheless, the difficulty of algorithms for event detection has limited most of the hate speech detection methods. Hate speech detection and classification with social media data has remain a vibrant research focus, but little research efforts have been devoted to the design of a generic architecture, threshold settings and fragmentation issues. Detection of hate speech from Twitter sources is based on techniques from different fields such as Machine Learning (ML), Natural Language Processing (NLP), data mining, extraction and retrieval of content, and text mining. However, Twitter streams contain large amounts of meaningless messages (Hurlock & Wilson, 2011), contaminated content (Lee et al., 2011), and rumours (Castillo et al., 2011), which adversely affect detection and classification algorithm performance.

In this study, a probabilistic clustering model for the classification of hate speech in twitter was developed in order to address issues with event detection in social media. The developed model is divided into four phases: metadata representation, training, clustering and classification. The metadata representation phase involve a generic metadata extractor for social media data collection. The training phase involves data pre-processing and labeling of hate tweets. Data preprocessing module is responsible for detecting hate tweets using the combinatorial algorithm based on Bayesian scoring function and labeling using crowdsourced of experts. Feature extractions using the Tri-gram and Wordngram models was performed at the data pre-processing stage to extract tweet features that will later serve as input in the classification phase. The clustering phase involves data representation module for topic modeling and detection module for real-time rule-based clustering of tweets into topic clusters. The data representation module is employed to create a topic model using probabilistic topic spotting measure based on Bayes theorem and score normalization technique. The idea of score normalization is to assign each tweet-topic similarity a unified threshold value between 0 and 1. Moreso, hate tweets database was used as training data for topic grouping to eliminate the problem of fragmentation associated with most online clustering. In addition to threshold setting and fragmentation issues, the data representation module was used to enhance the features extraction technique. Classification phase is responsible for hate speech classification based on semantic extracted features using fuzzy logic.

The main contribution of this study are: (1) The use of an automatic topic spotting measure based on naïve Bayes model to improve features representation (2) The design of a rule-based system for clustering real-time tweet into topic clusters based on a modified Jaccard similarity measure (3) The use of 4-level scale fuzzy model for hate speech classification.

In the remaining part of this work, Section 2 explore some background and related work. In Section 3, the designed model and algorithms are presented. Section 4 present the implementation, results and discussion of the developed model, and section 5 conclude the work.

2. Related work

Chau and Xu (2007) suggested a semi-automated approach to examine hyperlinks between web pages in order to identify populations with hate groups. The semi-automated approach to identifying hate group communities performs the extraction of user profile blog pages and determine their relationship with known hate group blogs. The authors then used clustering methods to perform further analyzes and determine the class of hate groups that a blog belongs to after the information about these blogs was collected. The authors also presented their findings graphically showing that their method can detect hate blogs.

Kwok & Wang (2013) proposed a text classification technique based on Naïve Bayes classifier for racist tweets. Their work is directly related to hate speech and it used labeled data to learn and build a Naïve Bayes classifier for the categorization between racist and non-racists tweets against black people. The approach focuses solely on unigram features and is therefore an application for text classification. The approach in this study relates to sentiment classification of hate tweets using the semantics of words (trigrams and wordngrams) by incorporating a lexicon of semantic terms and strength values developed using bootstrapping technique.

Warner & Hirschberg (2012) presented a supervised learning-based correlation that correctly categorizes hate speech as anti-Semitic, anti-Muslim and anti-African. They create an anti-Semitic-based language model and use correlation to identify the presence of hate speech in other categories. The study presented high accuracy rate but with low recall rate.

In a related study, Burnap & Williams (2015) developed an approach to classify hateful and antagonistic content on Twitter with a focus on race, ethnicity or religion; and more general responses. The results of their classification were optimized through the combination of Bayesian Logistic Regression, Random Forest Decision Tree and Support Vector Machine (SVM) with a voted ensemble *meta*-classifier. The study presented high accuracy rate but no generic architecture for hate speech classification.

Waseem & Hovy (2016) presented a predictive features for hate speech detection on Twitter using character n-gram based approach and Logistic Regression (LR) model. In conjunction with character n-grams for hate speech detection, they used LR to analyze the impact of different extra-linguistic features (gender, location and length). The study provide annotated hate speech dataset but with low detection rate.

Ribeiro et al. (2018) presented a user-centric view of hate speech detection using graph model. The graph model algorithm, takes advantage of the retweet network and outperforms content-based methods to hate speech detection. The limitation of the study is related to the issue of fragmentation and threshold settings.

Founta et al. (2019) suggested a single deep learning architecture using a wide range of available metadata such as followers, location, account age, total number of a user's posted / favored / liked tweets by a user. These metadata are then combined with hidden patterns which are automatically extracted in the tweet text to identify multiple highly interrelated abusive behavioral norms (hate speech, sexism vs. racism, bullying, sarcasm, etc.). The evaluation results showed an improvement over previous state-of-art methods.

Bellan & Strapparava (2018) presented a new set of features (grammar, space and character grams, bigrams and letter trigrams, character flooding or word exaggeration, level of education, relationships, semantic embedding, emotional features, etc.) to detect inappropriate news comments. The authors applied LR, Decision Tree and Naïve Bayes models to experimentally detect inappropriate comments. The results showed that LR outperforms the other models using the feature sets.

Greenwood et al. (2019) presented a NLP method to identify abusive language, the politicians it is targeted at, and the topics that tend to trigger abusive responses in the original tweet of the politician. To detect abusive language in tweets, the researchers used a dictionary-based approach. The model designed by the authors obtained slightly higher results than similar work on the same dataset.

In the work of Serra et al. (2017), a Multi-Layer Perceptron (MLP) was designed to perform hate speech categorization in Twitter data. Firstly, a language model for next-character prediction was trained with the data corresponding to each single category. Secondly, the error signal of class-based language models were used as input to the MLP classifier for hate speech categorization. The performance evaluation of the designed MLP classifier showed improved performances when compared to some state-of-the-art methods.

In a recent study, Siddiqua, Chy & Aono (2019) proposed a unified neural network model for detecting hate speech in Twitter. The unified neural network model make use of state-of-the-art pre-trained sentence embedding models for tweet feature representation. The first task of their proposed unified neural network model is to detect whether a tweet is hateful or not. Secondly, the task is to determine the severity level of the detected hate speech and identify targeted audience in reference to either individual or group of individuals. The proposed unified neural network showed better performances when compared to state-of-the-art methods.

In a related study, Setyadi, Nasrun & Setianingsih (2018) designed a backpropagation neural network for hate speech detection in Twitter messages. The performance of the designed backpropagation neural network was found to be average across all evaluation metrics.

In another study, Wiedemann, Ruppert & Biemann (2019) developed a neural network architecture for offensive language detection. It combines a Bi-directional Gated Recurrent Unit (GRU) layer (Cho et al., 2014) with a three parallel Convolutional Neural Network (CNN) layers, each with a different kernel sizes and a filter. The outputs of the three CNN blocks are converted into a single vector by global max-pooling. This vector is then fed into a final prediction layer for the detection of offensive language in Twitter messages with good prediction accuracy.

In a recent study by Corazza et al. (2018), the authors proposed a Recurrent Neural Network (RNN) architecture for detecting offensive language in Tweet messages. The proposed RNN architecture is designed to perform two tasks. Firstly, the RNN is designed to detect whether a tweet message is offensive or not offensive messages. Secondly, the RNN performs classification task to categorize the severity of offensive messages. The proposed RNN were optimized with word embeddings, emoji embeddings and social-network specific features representations. The RNN model is able to fused text-level features and tweet-level features in order to perform an improved classification tasks.

In another study, Pitsilis, Ramampiaro & Langseth (2018) proposed an ensemble of RNN classifiers for detecting offensive languages in Twitter data. The proposed approach includes different tweet-level features representation. The extracted features are fed as input to the ensemble RNN classifiers for final prediction. The evaluation results showed that the approach outperforms some state-of-the-arts algorithms. In addition, the approach was found to be efficient for racism, sexism and normal messages categorization.

Recently, Winter & Kern (2019) developed a CNN for multilingual hate speech detection on Twitter. They deployed CNN based on word embedding technique for feature extraction to detect hate speech either in English or Spanish from Twitter messages. Firstly, the task is to detect whether a tweet message contains hateful speech and secondly to determine the target and severity level of the hate speech. The developed CNN architecture produces better performances when compared to some baseline classifiers.

Similarly, Ribeiro & Silva (2019) proposed a CNN architecture using pre-trained word embeddings (GloVe and FastText) for hate speech detection against women and immigrants on Twitter. The developed CNN model for multilingual (English or in Spanish) hate speech detection consist of two classification tasks where the first task is to predict whether a multilingual tweet targeted against women or immigrants is hateful or not hateful, and second task is to decide whether the hate speech is targeted at a single individual or a group of individuals. The developed CNN architecture showed better performance in the Spanish tweets for hate speech detection.

In addition, Badlani, Asnani & Rai (2019) proposed a CNN model that first deployed words embedding techniques to extract features relating to sarcasm, humour, hate speech and sentiment in tweet messages. Secondly, the extracted features was then used as input into the CNN model for sentiment classification. The results of the developed CNN model showed better classification performance when compared to a model that only predicts sentiment.

The study of Florio et al. (2020) proposed the Bidirectional Encoder Representations from Transformers model pre-trained on Italian hate speech Twitter data (AlBERTo) for hate speech prediction. The AlBERTo model consist of an encoder layer for sentence-level feature representation and a decoder layer that produces a binary prediction for hate or non-hate speech. The AlBERTo model was found to outperform the linear SVM by using training data that has close temporal distance to the test set.

The study of Bosco et al. (2018) provide an overview of hate speech detection (HaSpeeDe) task on Italian social media (Facebook and Twitter) using different NLP models. The authors reported that Cimino et al. (2018) as one of the participants in the HaSpeeDe tested three independent classification models on both Facebook and Twitter data. The first one is based on linear SVM, another one based on a 1-layer Bidirectional Long Short Term Memory (BiLSTM) and the third one is based on a 2-layer BiLSTM. The results suggested that 2-layer BiLSTM performs better with Facebook data than with Twitter data overall, when compared to the other related systems with F1-score of 0.8288.

In another recent study, Polignano et al. (2019) proposed a fine-tuned AlBERTo model for hate speech detection on Italian Twitter data. The weights of the decoder layer of the traditional AlBERTo model is fine-tuned in order to predict correctly either a tweet is hate or non-hate speech using the testing set. The model was evaluated on data from both Facebook and Twitter domains according to four different tasks. The first two tasks involves training and testing the model on data from the same domain. On the other hand, the last two tasks involves training the model on data from one domain and testing it with data from the other domain. The results showed that the fine-tuned AlBERTo model performed better than related models, when tested on data from the same domain as the training data. On the other hand, the fine-tuned AlBERTo model showed reduced performance when tested on data from different domain as the training data.

Corazza et al. (2020) proposed a deep learning architecture for online hate speech detection on multilingual datasets consisting of English, Italian, and German languages. The authors used and compare three different recurrent architectures that include LSTM, GRU and BiLSTM. The recurrent architectures consist of the input layer, hidden layer and prediction layer. First, ngram models were used to extract word-level features and the input layer of the recurrent architectures was used to learn the encoded word-level features. Secondly, the learned encoded word-level features was then concatenated with the tweet-level features that includes emojis and emotion lexica as extracted in the recurrent input layer. The recurrent hidden layer used the extracted features for learning and training of the recurrent architectures. Finally, the output from the hidden layer was used as input for the prediction layer to predict hate or non-hate speech. The recurrent architectures were tested on the multilingual datasets and the best result was obtained using the LSTM model compared with the GRU and BiLSTM models. The results also showed that keeping the same features, BiLSTM model can achieve close results to the LSTM model.

In a more recent study, Paschalides et al. (2020) designed a three-layered deep learning method that monitors, detects and visualizes the incidence of hate-related speech using Twitter messages. The deep learning method includes CNNs and RNNs that automatically learn abstract feature representations from the input passing through the multiple weighted layers of the deep learning architecture for effective classification. The classification results showed better performances when compared to state-of-the-arts methods.

In another recent study, Huang, Xing, Dernoncourt & Paul (2020) presented a deep learning approach including CNN and RNN classifiers by using biGRU (Chung et al., 2014; Park et al., 2018) for hate speech identification in Twitter data. The results presented by the authors showed that the proposed deep learning methods outperformed LR classifier in the task of hate speech identification in Twitter messages.

In other study, Bisht, Singh, Bhadauria & Virmani (2020) developed a LSTM model for the detection of hate speech and offensive language in Twitter data. The developed LSTM model include a single LSTM layer consisting set of input layers that gets the consecutive input. The input layer then passes the data to the LSTM hidden layer for the sequence hate-related input data. The data was then dispatched to the classification layer for the final categorization between hate speech and offensive language in Twitter messages. Finally, the developed LSTM model showed that it can outperform other baseline models for hate speech classification.

Table 1 provide a collection of hate speech benchmark datasets that can be used by researchers to test their developed methods and other ML models in the future research of hate speech classification.

2.1. Social media

It is described as an Internet based application that enables users to create, access and share remotely accessible digital content (Kaplan & Haenlein, 2010). In other words, it consists of big data generated by users by means of short messaging. For the processing and evaluation of these big data, most research models are not standardized and are therefore limited to a single social media site for event detection. The challenge of designing a standardized architecture is inherent in the unstructured and dissimilar nature of data generated from different social media sites. Social media sites are made up of business-based networking channels (LinkedIn), location-based (Foursquare), content sharing (Facebook, PInterest, Blogs), photo sharing (Flickr, Instagram), microblogging (Twitter), video platforms (Youtube, Vimeo) etc., (Mainka et al., 2014). As a source of information and as a super set for microblogging, social networking platforms contain uncertain and user-generated content (Westerman et al., 2012; Nip & Fu, 2016). Fig. 1 shows some of the popular network icons for social media.

2.2. Hate speech detection

Hate speech is commonly defined as any message that appears offensive to a person or group based on specific characteristics such as colour, race, gender, sexual orientation, nationality and religion (Nockleby, 2000). Social media environment, and particularly twitter, provides a fertile ground for producing, posting, and exchanging hate messages against a perceived enemy group. Analysis of hate speech and feelings is closely related. Previous works have attempted to detect negative and positive feelings from wordlists generated from twitter datasets (Dang, Zhang & Chen, 2009; Pang & Lee, 2008). Wordlist opinion generation methods fall into two main categories, dictionary and corpus-based approaches.

The dictionary-based approach includes a fixed dictionary of semantically relevant words marked with both a polarity label and a semantic orientation score (Taboada, Anthony & Voll, 2006). The dictionary is initially developed using a bootstrapping strategy with a small set of seed opinion terms and an online dictionary like WordNet (Miller

Fig. 1. Popular social media network icons.
Source: Howells and Ertugan (2017).

Table 1
Benchmark datasets for hate speech classification in Twitter.

Language	Dataset name	Annotation type	Size	Data link	Reference
Arabic	L-HSAB	Hate speech and abusive language	5846	github.com/Hala-Mulki/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset	Mulki, Haddad, Ali & Alshabani (2019)
Danish	DKhate	Offensive speech, target, and grade	3600	N/A	N/A
English	Davidson et al.	Hate speech and offense	25,000	github.com/t-davidson/hate-speech-and-offensive-language	Davidson et al. (2017)
	Wikipedia Detox	Personal attacks	100,000	figshare.com/articles/Wikipedia_Detox_Data/4054689	Wulczyn, Thain & Dixon (2017)
	Waseem & Hovy	Hate speech	16,000	github.com/zeerakw/hatespeech	Waseem & Hovy (2016)
	OffensEval 2019	Offensive speech, target, and grade	14,000	competitions.codalab.org/competitions/20011	Zampieri et al. (2019)
	Liu et al.	Hostility	30,000	N/A	Liu, Guberman, Hemphill & Culotta (2018)
	StormfrontWS	Hate Speech	10,568	github.com/aitor-garcia-p/hate-speech-dataset	de Gibert, Perez, García-Pablos & Cuadros (2018)
	hatEval	Targeted hate speech; aggressive behaviour	13,000 English 6600 Spanish	competitions.codalab.org/competitions/19935	i-Orts (2019)
	Founta et al.	Hateful and Abusive speech	100,000	https://github.com/ENCASEH2020/hatespeech-twitter	Founta et al. (2018)
Indonesian	Ibrohim & Budi	Hate speech & abusive language	13,169 docs	github.com/okkyibrohim/id-multi-label-hate-speech-and-abusive-language-detection	Ibrohim & Budi (2019)
German	GermEval 2018	Offensive language	8541	github.com/uds-lsv/GermEval-2018-Data	Wiegand, Siegel & Ruppenhofer (2018)
Spanish	hatEval	Targeted hate speech; aggressive behaviour	13,000 English 6600 Spanish	competitions.codalab.org/competitions/19935	i-Orts (2019)
Portuguese	Fortuna et al.	Hate speech	5668	github.com/paulafortuna/Portuguese-Hate-Speech-Dataset	Fortuna, da Silva, Wanner & Nunes (2019)
Italian	HSC	Hate speech	6000	github.com/msang/hate-speech-corpus	Sanguinetti, Poletto, Bosco, Patti & Stranisci (2018)
Polish	PolEval 2019	Cyberbullying & hate speech	11,135	poleval.pl/tasks/task6	Ptaszynski, Pieciukiewicz & Dybała (2019)

et al., 1990) and SentiWordNet (Esuli & Sebastiani, 2006) in a number of the proposed methods. Generally speaking, dictionary-based approaches suffer from the inability to find opinion terms with different subject and meaning orientations.

On the other hand, corpus-based approach uses a domain corpus with a chosen syntactic or co-occurrence pattern to capture opinion terms (Araújo et al., 2020). Using the rule-based methods of natural language analysis, syntactic, structural and sentence level features are used to evaluate the linguistic nature of words and phrases to be included in a lexicon of opinion (Chang, Ku & Chen, 2019). Through this approach, a lexicon is filled with words and phrases that are more suited to the field by adding contextual features that could potentially change an opinion word's semantic orientation.

2.3. Sentiment analysis

Analysis of sentiment aims to identify and classify opinions, emotions, and attitudes contained in tweets as positive, neutral, or negative in meaning. In sentiment classification, most researchers use classifiers such as SVM and Naïve Bayes, but these classifiers cannot really capture certain emotions that can overlap between positive or negative classification (Sakaki et al., 2010). Therefore, the advantage of using fuzzy logic for sentiment analysis is to capture contrasting values between positive or negative classification.

2.4. Combinatorial algorithm

The concept of a combinatorial pattern matching algorithm emerged from the particular bioinformatics problem of searching for a known sequence in a sequence database (Folorunso, Ayo & Babalola, 2016). Khreich et al. (2009) claimed that event detection could be addressed similar to anomaly detection techniques that rely on modeling normal user behavior and detecting any deviation from this baseline profile. This study uses the pattern matching algorithm method to distinguish hate case tweets from the sampling of all the tweets posted by all Twitter users. As inputs, the combinatorial algorithm takes ordinary user database which normally tweet about hate and unknown tweets database of all the tweets posted by all Twitter users. Consequently, any divergence from the users' regular list that normally tweet on hate reflects non-hate tweets as shown in Fig. 2.

2.5. Bayesian scoring function (BSF)

BSF is a variant of Bayesian network which is a probabilistic network that can be applied to various fields such as detection, prediction, ML, finance and robotics (Tian et al., 2013). Bayesian network is a probability distribution in the context of this study, which can be used to distinguish a single hate tweet from all other tweets based on the score returned by the combinatorial algorithm. The notion of degree of support in Bayesian network was adapted to classify tweet item as hate tweet otherwise as non-hate tweet based on a specified threshold value as in Eq. (1).

$$S(A \cup B) = \frac{x}{n} \tag{1}$$

where $A \cap B = \varnothing$, and $\times$ is the number of tweets containing all the A and B tweet items; n is the total number of a corpus or database tweets, and A & B are two separate events.

2.6. Online clustering

The focus of online clustering is to group event tweets automatically into sets of tweets, called clusters, so that each cluster contains tweets related to a particular topic. This issue, called topic detection, is similar to document domain clustering and differs from the classification of tweets (Bello-Orgaz, Jung & Camacho, 2016). An incoming event tweet is permanently assigned to the most similar event clusters based on a similarity measure and a pre-defined threshold. An efficient clustering algorithm (Curiskis, Drake, Osborn & Kennedy, 2020), will take into consideration the element of time distance and content for clustering. A clustering algorithm should be sufficiently modified to work online and become more robust to noise. Online clustering algorithm maintains an active cluster list. A linked function vector (i.e. weighted list of keywords) is collected from the terms of the embedded tweets in each clusters and weighted using the TF-IDF measure (Salton & Yu, 1973). Each cluster's time centroid is stored, which is the mean time of all the cluster tweets being published. If the time centroid is greater than a defined number of days, a cluster will be labeled inactive, in which case no additional tweet can be added to the cluster. Using TF-IDF, an input event tweet is first represented by its feature vector representation. Then, to calculate the distance between the tweet and a candidate cluster c, a similarity measure is required (Steinbach et al., 2000). So long as its range is less than or equal to a pre-specified constant, a tweet is added to the nearest cluster (iterative clustering). If there is no such cluster, a new cluster (incremental clustering) will be started with the tweet as its only member.

2.7. Fuzzy logic

It can be defined as a multi-valued logic in which the truth values involves all intermediate values between 0 and 1 both inclusive (Zadeh, 1965). In most classification systems, fuzzy logic has turned out to be a popular concept to resolve uncertainty and for precision classification.

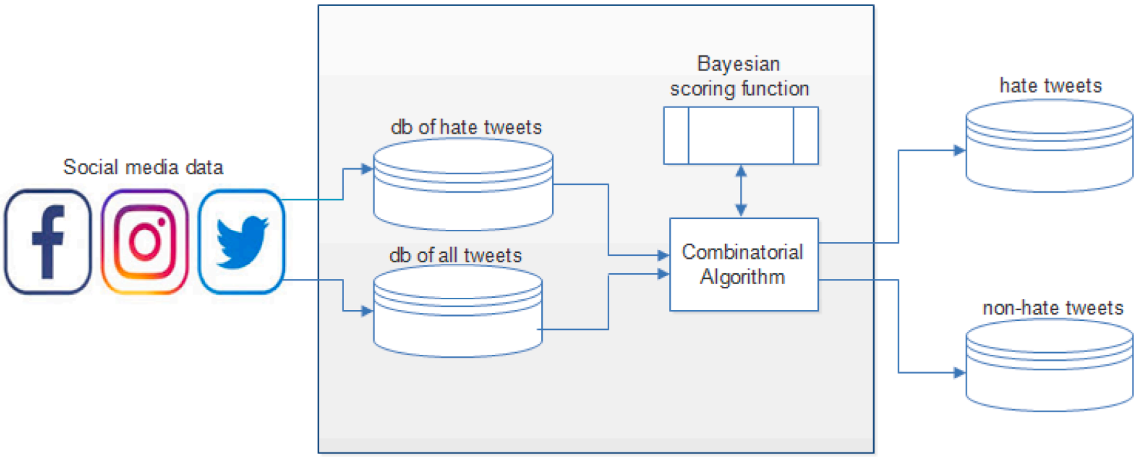

Fig. 2. Hate tweet filtering based on combinatorial algorithm.

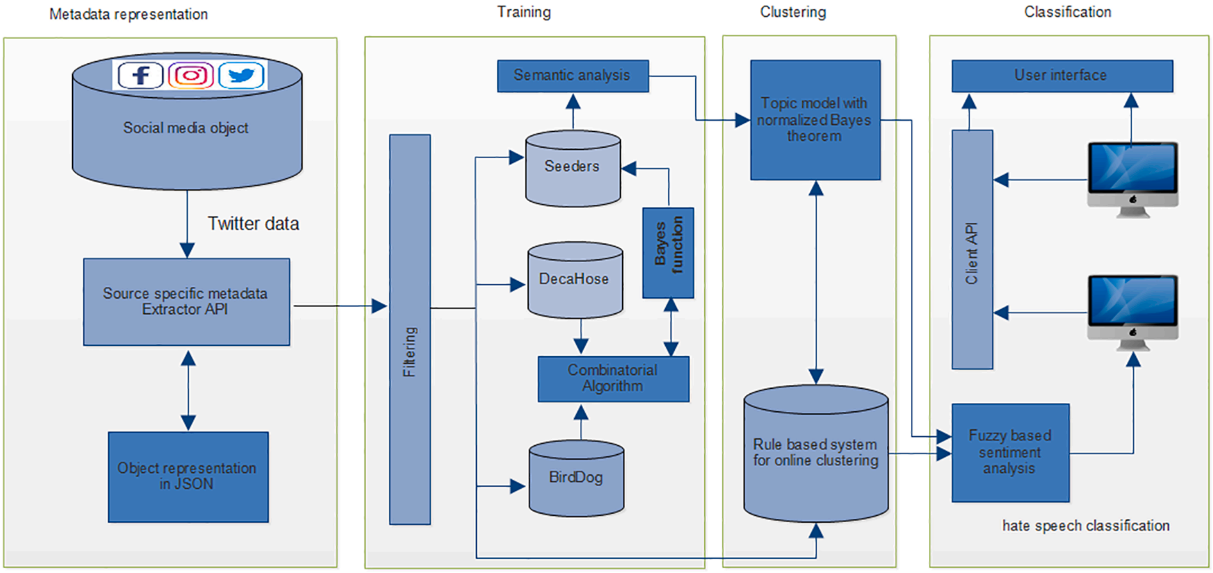
Metadata representation
Training
Clustering
Classification
Social media object
Twitter data
Source specific metadata Extractor API
Object representation in JSON
Filtering
Semantic analysis
Seeders
Bayes function
DecaHose
Combinatorial Algorithm
BirdDog
Topic model with normalized Bayes theorem
Rule based system for online clustering
User interface
Client API
Fuzzy based sentiment analysis
hate speech classification

Fig. 3. A probabilistic clustering model for hate speech classification in twitter.

Hence, fuzzy logic concept is most suitable for regulating instability in an unstable system. The concept was first suggested by Zadeh (1965) of the University of California at Berkeley. In a related study, Zadeh (1975) expounded on his own ideas of linguistic variables also called fuzzy sets. The comprehensive concept and basic components of fuzzy logic can be found in (Fullér, Hassanein & Ali, 1996; Wang, 1994). The basic components includes fuzzy sets, membership function, fuzzification, fuzzy range, inference engine and defuzzification.

3. Research methodology

In this section, the details of the proposed probabilistic clustering model for hate speech classification in twitter is discussed. The method proposed is divided into four phases: metadata representation, training, clustering and classification as shown in Fig. 3. The metadata representation involve a generic metadata extractor api for social media data collection from different domains. The training phase involves the filtering method, combinatorial algorithm, BSF and semantic analysis for data pre-processing. The filtering method is to separate noise from the incoming tweets. Moreso, the purpose of the data pre-processing module is to separate hate speech from non-hate speech tweets using combinatorial algorithm based on a normalized threshold value using BSF. The reason for the choice of combinatorial algorithm is because it is suitable for text similarity and general online detections. Semantic analysis is also employed at the training phase to extract tweet features that will later serve as input in the classification module. The clustering phase involves data representation and detection modules. The data representation module is employed to automatically create topic model using probabilistic topic spotting measure based on Bayes theorem. The topic model is built using database of hate speech from the training phase as training stories. The database of hate speech was used as training data for topic grouping to eliminate the problem of fragmentation associated with most online clustering. The detection module is a rule-based system developed for clustering real-time tweet into any of the created topic models based on a modified Jaccard similarity measure. The rule-based detection system help in the switching between iterative and incremental clustering for the detection of real-time hate tweet using a score normalization technique. The score normalization technique is to enable a unified threshold value for every tweet-topic similarity. Lastly, the classification phase include the classification module responsible for hate speech classification based on features extracted in the training phase and user interface for system interaction. The classification module uses fuzzy logic for a 4-level sentiment classification of hate speech. The details of the components in the proposed architecture (see Fig. 3) is highlighted below:

3.1. *Metadata representation phase*

3.1.1. *Source specific metadata extractor API*

The Twitter Streaming API (Davidson et al., 2017) was used to randomly collect the datasets used in the evaluation through the DecaHose and BirdDog access levels. [1]The DecaHose is a twitter streaming API that consist of 10% of all tweets posted by all Twitter users. [2]The BirdDog is the term used in this study to denote filtered streams consisting of tweets with hate-related lexicons. The collected datasets are made up of Twitter public live streams. A random sample of 25,000 English tweets containing 3639 terms from the hate speech lexicon compiled by [3]Hatebase.org, were extracted through the BirdDog API and was manually labeled as one of two categories by experts: hate speech and non-hate speech. A total of 5, 810 tweets end up in the hate speech category while 19, 190 tweets end up in the non-hate speech category. The experts compared the terms in the tweets with those hate speech lexicons provided by Hatebase.org and labeled accordingly. The category of labeled hate tweets was then collected in the database and used as input to the combinatorial algorithm representing normal database of hate tweets. While the DecaHose comprises 10% of unlabeled tweets posted by all Twitter users. The DecaHose is simply checked against the BirdDog database. The proposed probabilistic clustering model was developed to predict whether a tweet in English is hateful or not hateful.

3.1.2. *Object representation in JSON*

This is to convert any form of social media dataset into a format specified and required by the system architecture. Metadata Representation Structured Model (MRSM) is employed for generic representation of the social media metadata. MRSM is based on the JavaScript Object Notation (JSON) model describing social media object in terms of attribute–value pairs and array data types. The algorithm for the MRSM

[1] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/sample-realtime/api-reference/decahose

[2] https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/api-reference/get-tweets-search-stream-rules

[3] https://www.hatebase.org/

is shown in Appendix I. Hence, MRSM is defined as a structured text file consisting of three composite attributes: temporal (time), spatial (location) and semantic, describing every social media object:

- **Temporal attribute**

One of the three main attributes used to describe social media objects using MRSM is their temporal coverage. Temporal coverage consists of a set of timestamps, describing the duration or capture of an object (or event).

- **Spatial attribute**

The spatial attribute describe the surface coverage or location in which a social media object is created or in which an event occurs.

- **Semantic attribute**

The semantic attribute is used to describe the terms and relationship in a social media object (or event). It describe a set words/expressions having identical semantic meanings, and a set of links/semantic relations connecting the words.

3.2. Training phase

The data extracted from the metadata representation phase serve as input to the training phase. The training phase houses the pre-processing module which consists of the filtering method, combinatorial algorithm, BSF and semantic analysis. The details of each component in the training phase is highlighted below:

3.2.1. Filtering method

The filtering method was used to separate noise from the extracted tweets in the metadata representation phase. The resultant noise free tweets are the DecaHose and BirdDog databases from the source specific metadata extractor API. See Appendix II for the scoring algorithm describing the entire training phase. The tweets contained in the DecaHose and BirdDog databases are pre-processed by the following basic information retrieval tasks:

i. Stopwords removal: It is used to remove article, preposition and conjunction in the collected tweets. Removing stopwords will not just help the system minimize noise, it can also reduce system processing time.
ii. URL removal: It is used to remove URL and useless links in the collected tweets.
iii. Others include removing blank comments and non-ASCII characters in the collected tweets.

3.2.2. Combinatorial algorithm

The pre-processed tweets in the DecaHose and BirdDog databases serve as input to the combinatorial algorithm. The DecaHose and BirdDog databases represents unknown tweets and hate tweets profiles respectively. In order to extract more hate tweets, the combinatorial algorithm used the Jaccard similarity index (Eq. (2)) to compute the distance scores between an unknown tweet (D) and the hate tweets (B) profiles.

$$J(D,B) = \frac{D \cap B}{D \cup B} \tag{2}$$

$$0 \leq J(D,B) \leq 1; D, B \neq \varnothing$$

3.2.3. Bayesian scoring function (BSF)

The BSF that is adapted in this study is the degree of support. The degree of support denoted as $S(D \cup B)$ between two instances D⇒B, is the ratio of number of tweets D, in B to the total number of B as shown in Eq. (3).

$$S(D \cup B) = \frac{nooftweetD, inB}{totalnoofB} = \frac{J(D,B)}{B} \tag{3}$$

where$J(D,B)$, is the score returned in Eq. (2) and B is the total number of hate tweet items in BirdDog database. If $S(D \cup B)$ is greater than or equal to a normalized threshold, then a tweet item is classified as hate tweet and stored in the seeders database, otherwise as non-hate tweet as shown in Eq. (4).

$$seeders = \begin{cases} hatetweetifS(D \cup B) \geq nThreshold \\ \quad nonhatetweet \end{cases} \tag{4}$$

The *nThreshold* stands for normalized threshold as defined in Eq. (5). This threshold value is adaptable and thus, can eliminate misclassification.

$$nThreshold = \frac{J(D,B) - \mu}{\sigma} \tag{5}$$

where $J(D,B)$represents the tweet-tweet similarity measure, μ is the overall mean for tweets scores in B and σ the overall standard deviations for tweets scores in B. The parameters μ and σ are defined in Eqs. (6) and (7) respectively.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

$$\sigma = \frac{1}{N}\sum_{i=1}^{n} (x_i - \mu)^2 \tag{7}$$

where N = size of B, n = size of D andx_i = Jaccard score between a given tweet in D and tweets in B.

3.2.4. Semantic analysis

Sentiment analysis techniques are used to perform opinions classification. It is debatable that this opinions classification technique can be employed to perform hate speech classification. However, in the case of hate speech, some texts might portray negative meaning or might even be hate-related, but the context makes them not hate speech- related. Hence, the need for method to detect both the individual and context meaning of text. The purpose of semantic analysis at the training phase is to extract semantic features, such as emoticons, hashtags and text from pre-processed hate tweets in the seeders database. These features are expected as input for sentiment classification in the classification phase. This study uses tri-grams and wordngrams to create bag-of-words from the pre-processed hate tweets in the seeders database. The bag-of-words created are then compared with corpus-based generated opinion lexicons for semantic features and their scores. Table 2 shows sample emoticons, text and hash tags with their respective scores. The bag-of-words model (Jones, 1972; Harris, 1954) are wordgrams commonly used in text classification where the frequency of occurrence of each word was used as a feature for training a classification algorithm. The tri-gram and wordngram are variants of the TF-IDF model used to create the bag-of-words. The following are definitions of the opinion features as extracted by tri-gram and wordngram models:

- **Emoticons**: These are symbolic depictions of facial expressions using punctuation and letters. In this study, the semantic scores ranges between 0 and 1. Opinions that relates to hate speech will tends to 1 and non-hate speech will tends to 0. Semantic score is the orientation score based on the subjectivity and semantic features of each hate lexicons in a text (Gitari et al., 2015).
- **Text:** These are word, phrase and abbreviations depicting user mood. These include adverbs, verbs, adjectives and text abbreviations such as laughing out loud (LOL) etc.

Table 2
Sample features.

Emoticon	Meaning	Score	Verb	Score	Adverb	Score	Adjective	Sore
:D	Big grin	0.1	Love	0.1	Complete	0.1	Sarcastic	0.2
XD	Laughing	0.1	adore	0.1	most	0.2	narcissistic	0.2
\m/	Hi 5	0.1	like	0.2	totally	0.5	gay	0.8
:), =), :-)	Happy, smile	0.2	enjoy	0.2	extremely	0.5	guilty	0.3
:*	Kiss	0.2	smile	0.2	too	0.2	obnoxious	0.2
;)	Wink	0.2	impress	0.1	very	0.1	disgusted	0.3
:|	Straight face	0	attract	0.1	pretty	0.1	anxious	0.4
:\	Undecided	0	excite	0.1	more	0.2	foolish	0.4
:(	Sad	0.75	fuck	0.7	anyway	0.1	irritated	0.5
:P	Tongue	0.75	reject	0.2	any	0.2	sick	0.7
</3	Broken heart	0.75	disgust	0.3	quite	0.3	hoe	0.8
B(	Sad with glasses	0.75	blacklisted	0.6	little	0.4	disgruntled	0.9
:'(	Crying	0.9	dislike	0.7	less	0.6	nasty	0.9
X-(	Angry	0.9	detest	0.8	not	0.8	annoyed	0.9
>:(	Grumpy	0.9	suck	0.9	never	0.9	downcast	0.9
3:)	Devil	0.9	hate	0.95	hardly	0.95	bitter	0.9

-

Hashtags: Users use hashtags "#" to mark topics. It is used by Twitter users to make their tweets visible to a greater audience. This hashtags could fall in the text group of verbs, adverbs, adjectives and text abbreviations.

3.2.5. *Scoring module*

The scoring module is divided- into three scores: Text group, emoji group and hashtag group. The individual group score being normalized between 0 and 1. The scoring module identifies the emoji, adjective, verb and adverb terms from the bags of words created in the semantic analysis stage. The scoring module discards the preceding and succeeding words occurring with the emoji, adjective, verb and adverb in the tweets. Then, the adjective, verb and adverb terms are categorized into emoticon, text and hashtag groups. For example, the scoring module will extract the adjective "fucking" from the wordngram "you're fucking" by discarding "you're" from it and categorize it under the text group. The scoring module then compute the semantic score of the emoticon, text and hashtag groups in a given tweet using their respective semantic score tables. These feature groups and their scores will later serve as input into the fuzzy logic for the overall sentiment classification of the given tweet. The following Equations define the formula for computing the semantic score for the text, emoji and hashtag features in a given tweet:

- **Text group score**

$$S(TG_i) = \bigcup_{i=1}^{n} S(verb_i) \tag{8}$$

where,$S(TG_i)$ denote the text group score, $S(verb_i)$denotes the maximum score of the ith verb or adjective or adverb group.

- **Emoji** group **score**

$$S(EG_i) = \sum N_{ei} * S(E_i) \tag{9}$$

where, N_{ei} denotes the count of the ith emoticon and S(Ei) denotes the score of the ith emoticon.

- **Hashtag** group **score**

$$S(HG_i) = \sum N_h * S(H_i) \tag{10}$$

where, N_h denotes the count of the ith hashtags and S(Hi) denotes the score of the ith hashtag.

The following running example illustrates the computation process of the scoring module:

Example 1:. *Given the tweet "you're fucking gay, blacklisted hoe holding out for #TehGodClan anyway", the keywords in the example tweet include: fucking (adjective), gay (adjective), hoe (adjective), blacklisted (verb), #TehGodClan (hashtag), anyway (adverb). Therefore, Eqs.* (8)–(10) *are computed as follows in reference to* Table 2 *for the respective semantic scores of the keywords:*

- Text group score

$$S(TG_i) = \bigcup_{i=1}^{n} S(verb_i)$$

$S(TG_i)$ = S(fucking) $\bigcup S$(gay) $\bigcup S$(hoe) $\bigcup S$(blacklisted)$\bigcup S$(anyway) = 0.7 $\bigcup$0.8 $\bigcup$0.8 $\bigcup$0.6 $\bigcup$0.1 = 0.8

where $S(TG_i)$ returns the maximum score of all the union of adjectives, verbs and adverbs in the sample tweet.

- Hashtag group score

$$S(HG_i) = \sum N_h * S(H_i)$$

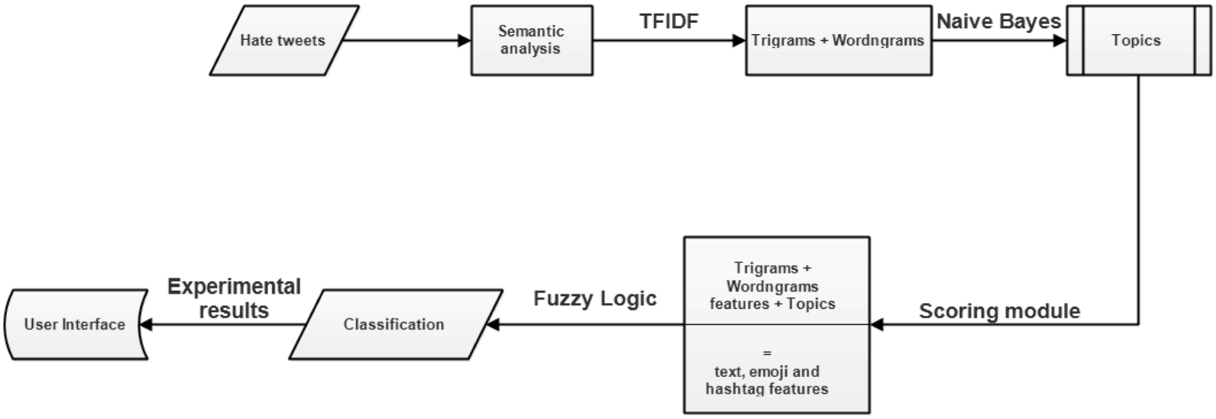

Fig. 4. The combination of features to improve features representation.

H_i = TehGodClan, N_h = no of times TehGodClan appear in the tweet and $S(H_i)$ = score of the hashtag

Therefore, $S(HG_i) = N_h \times$ S(TehGodClan) = 1 × 0.5 = 0.5

- Since there is no emoji, we assumed a score of 0.5 for the emoticon group

Hence, the feature groups (text, hashtag and emoji) and their computed scores will serve as input to the fuzzy logic in the classification of a given hate tweet according to a defined rule combinations. Each of the feature group score representing some linguistic variables defined in a fuzzy value range. For instance, the classification module based on a defined rule combination will compute the sentiment class for the example tweet as follows:

IF (emoji is 0.5) and (hashtag is 0.5) and (text is 0.8) THEN class is MODERATELY SEVERE OFFENSIVE.

3.3. Clustering phase

The clustering phase consist of the data representation and the rule-based clustering system for automatic topic grouping and online clustering respectively.

3.3.1. Data representation

The data representation module is employed to automatically create topic clusters using probabilistic topic spotting measure (M_{TS}) based on naïve Bayes classifier (Schwartz et al., 1997) as shown in Eq. (11). The topic model was built using database of seeders from the training phase as training stories. The seeders was used as training data for automatic topic grouping to eliminate the problem of fragmentation associated with most online clustering. In addition to eliminating the problem of fragmentation, the purpose of the naïve Bayes for automatic topic modelling is to improve the performance of the features representation by the semantic analysis for hate speech classification. See Appendix III for data representation algorithm.

In order to manage the problem of imbalance data and overfitting in classification and strengthen the simulation results, k-fold (k = 5) cross-validation was used. The training data in the seeders database was randomly divided into 5 equal sized subsets of the dataset. A single subset was used to test the proposed rule-based clustering method and the remaining 4 subsets were used as training data for the naïve Bayes classifier.

$$M_{TS}(S,T) = log\frac{P(T\backslash S)}{P(T)} = log\frac{P(S\backslash T)}{P(S)} \quad (11)$$

where P(T\S) is the likelihood that T is the topic provided all the terms in tweet, P(T) is the first likelihood that T is the topic, and vice versa.

Fig. 4 shows the details of the proposed features representation method for hate speech classification in Twitter. First, the labeled hate tweets in the seeders database are transformed into features vector with the TF-IDF (Trigrams and Wordngrams) features extraction technique. In order to improve the performance of the TF-IDF features representation, naïve Bayes model was used to automatically infer topics from the tweets. A scoring module is then used to combine the features extracted by both the TF-IDF and the naïve Bayes models. The scoring module group the combined features into text, emoji and hashtag for input into the fuzzy logic classification model. The fuzzy logic used the input features and their computed scores to categorize a hate tweet into either one of mildly offensive, moderately offensive, moderately severe offensive and severely offensive. Finally, the user interface was used to establish interaction between users and the system for further interpretation on the validity of experimental results.

3.3.2. Rule-based clustering

The rule-based clustering assigns a given tweet to any of the topic clusters created in the data representation module based on a modified Jaccard similarity measure in Eq. (12). The rule-based clustering system was developed to help in the switching between iterative and incremental clustering for the detection of real-time hate tweet. The rule-based system permit iterative clustering i.e., a new tweet is added to one of the topic clusters if the Jaccard similarity measure is less than or equal to a normalized score shown in Eq. (13). On the other hand, the rule-based system permit incremental clustering i.e., the new tweet is used to form a new topic cluster if the Jaccard similarity measure is greater than a normalized score. The idea of score normalization is to assign a unified threshold value between 0 and 1 for every tweet-topic similarity. Furthermore, a cluster C is considered inactive and removed from the list of currently active clusters, if no new tweet item is added to it during its life-span. See Appendix IV for the rule-based clustering algorithm. The justification for using clustering method is due to its suitability for automatic tweets classification into topic clusters (Curiskis, Drake, Osborn & Kennedy, 2020). Similarly, the reason for Jaccard similarity is because of its popularity for keywords similarity in text analysis, as repetition of a word does not reduce their similarity unlike other distance measures (Niwattanakul et al., 2013).

$$J_{sim}(tw, CC) = \frac{\sum_{j=1}^{I^i} sgn(i) \cap sgn_C.ff(j)}{\sum_{j=1}^{U^i} sgn(i) \cup sgn_C.ff(j)} \times f(t_c) \quad (12)$$

where $sgn(i)$ denotes the new tweet signature, $sgn_C.ff(j)$ denotes the signature for a cluster, $I^i = sgn(i) \cap sgn_C.ff(j)$and $U^i = sgn(i) \cup sgn_C.ff(j)$ are the sum of frequencies of the terms appearing in the intersection and union between the new tweet and all the tweets in a given cluster, respectively and $f(t_c)$ denotes the cluster fading function as shown in Eq. (14). The fading function is a time-dependent function that diminishes the importance of a cluster.

$$score_N(tw, CC) = \frac{J_{sim}(tw, CC) - \mu}{\sigma} \tag{13}$$

where $J_{sim}(tw, CC)$ represents the tweet-cluster similarity measure as defined in Eq. (12), μ is the overall mean for topic clusters and σ the overall standard deviations for topic clusters.

Example 2:. *For instance, consider the tweet of example* 1, *and suppose that we have clusters with the following word unigrams appearing in all the tweets contained in the clusters,* w_u^C*={kate, resentful, little, fucking, gay, hoe, suck, detest, love, henshaw, ugly, killed}, and corresponding frequencies* ff_{w_u}={8,7,10,4,5,4,10,8,3,8,3,1}, *then the mean* μ *between the word unigrams of the tweet in example* 1 *and the clusters is given by* $\{1 \times 8 + 1 \times 7 + 1 \times 10 + 1 \times 4 + 1 \times 5 + 1 \times 4 + 1 \times 10 + 1 \times 8 + 1 \times 3 + 1 \times 8 + 1 \times 3 + 1 \times 1 = 71\}$ *and the standard deviation* σ *can be computed similar to* Eq. (7).

$$f(t_c) = 2^{-\lambda(t_o - t_c)} \tag{14}$$

where t_0 is the creation time of the cluster and t_c is the time of the last item assigned to C. The decay rateλ of a cluster is defined as shown in Eq. (15). The decay rate of a cluster denote the half-life of the cluster or the rate at which the importance of the cluster diminishes when no new tweet item is added to it for a specified period of time.

$$\lambda = 1/ls \tag{15}$$

where ls is the life span of a cluster as defined in Eq. (16). The life span of a cluster is the time duration a cluster is retained before it is removed from the list of active clusters. The parametersδ and h in Eq. (16) are user defined constraints. δ is the minimum time duration a cluster is retained even if no new tweet is added to it, while h, regulates the time-based distance a cluster is considered *active*.

$$ls = \delta + 2h \times (t_c - t_0) \tag{16}$$

A cluster C is considered *inactive*, and removed from the list of currently active clusters, if no new tweet item arrives during its life span. This means that,$t_o - t_c \geq ls$, that is, $f(t_c) \leq 0.5$.

The following running examples explain the definitions of each components in Eq. (12) and illustrate its computation.

Example 3.. *Given the tweet of example* 1 *"you're fucking gay, blacklisted hoe holding out for #TehGodClan anyway", in order to decide which of the topic clusters created in the data representation module the given tweet belong, the following notions and definitions holds*:

Definition 1:. *A tweet, tw posted from a location l by a user u at the time t, can be represented as a tuple, denoted as,* $tw = (o, u, t, l, sgn)$, *where o is the object identifier and* $sgn = (w_u, w_b, h_u, h_b, m_u, m_b)$ *is a feature vector, known as signature. Each feature is defined as a list of items as follows*:

i. w_u are the words appearing in the tweet;
ii. w_b are the word bigrams;
iii. h_u are the hashtags appearing in tweet;
iv. h_b are hashtag bigrams;
v. m_u are the mentions contained in the tweet;
vi. m_b are mention bigrams.

The tweet in example 1 "*you're fucking gay, blacklisted hoe holding out for #TehGodClan anyway*", is a sample hate tweet waiting to be assigned to any of the created topic clusters. The signature, $sgn(i)$ of this tweet is w_u= {fucking, gay, blacklisted, hoe, holding, out, love, anyway}, w_b= {fucking gay, gay blacklisted, blacklisted hoe, hoe holding, holding out, out love, love anyway}, h_u= {#TehGodClan}, $h_b = m_u = m_b = \varnothing$.

Definition 2:. *The centroid CC of a cluster C is a tuple* $CC = (c, t_o, t_c, sgn_C)$, *where c is the cluster label,* t_o *is the creation time of the cluster,* t_c *is the time stamp of the last time a tweet item was added to C, and* $sgn_C = sgn.ff$. *sgn is the feature vector similar to the tweet signature, while* $ff = (f_{wu}, f_{wb}, f_{hu}, f_{hb}, f_{mu}, f_{mb})$ *is the list of frequencies corresponding to the signature. Notice that the difference between the signature of a tweet,* $sgn = (w_u, w_b, h_u, h_b, m_u, m_b)$ *and that of a centroid,* $sgn_C.sgn$=$(w_u^C, w_b^C, h_u^C, h_b^C, m_u^C, m_b^C)$ *is that each feature in the former contains the list of items appearing in the tweet, while for the latter it is the list of items occurring in all the tweets assigned to C without repetitions.*

Example 4:. *Consider the tweet of example* 1, *and suppose that we have a cluster with the following word unigrams appearing in all the tweets contained in the cluster,* w_u^C*={kate, resentful, little, fucking, gay, hoe, suck, detest, love, henshaw, ugly, killed}, and corresponding frequencies* ff_{w_u}={8,7,10,4,5,4,10,8,3,8,3,1}, *then the intersection between the word unigrams of the tweet in example* 1 *and this cluster is given by* I^{w_u}=$sgn(i) \cap sgn_C.ff(1)$ = *{fucking, gay, hoe, love}, and their union is given by* U^{w_u} =$sgn(i) \cup sgn_C.ff(1)$ = *{ kate, resentful, little, fucking, gay, hoe, suck, detest, love, henshaw, ugly, killed, blacklisted, holding, out, anyway}. Then* $sgn(i) \cap sgn_C.ff(1)$= $\{4 + 5 + 4 + 3 = 16\}$, *while* $sgn(i) \cup sgn_C.ff(1) = \{8 + 7 + 10 + 4 + 5 + 4 + 10 + 8 + 3 + 8 + 3 + 1 + 1 + 1 + 1 + 1 = 75\}$. *The other term* $f(t_c)$ *of* Eq. (12) *is computed by the difference between the*

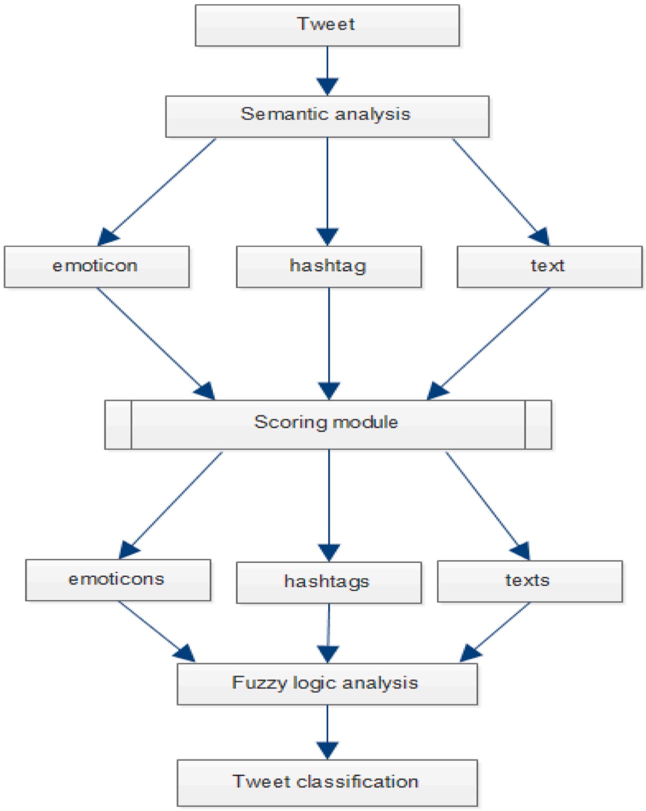

Fig. 5. Fuzzy-based sentiment classification.

Table 3
Fuzzy value range.

Linguistic value	Value range
Mildly	$0.1 \leq x < 0.3$
Moderately	$0.3 \leq x < 0.6$
Moderately severe	$0.6 \leq x < 0.8$
Severely	$0.8 \leq x \leq 1.0$

creation time and the last time a tweet is added to the cluster.

3.4. Classification phase

The classification phase consist of the classification module for hate speech classification and the user interface for system interactions. See Appendix V for sentiment classification algorithm.

3.4.1. Classification module

Fig. 5 shows the process of the classification module. The classification module is based on a fuzzy logic with the features extracted for sentiment analysis in the training phase as input. Sentiment analysis intends to identify opinions, emotions, and attitudes contained in tweets and classify them into positive, neutral or negative classes. Most studies used classifiers such as SVM and Naïve Bayes for sentiment classification, but these classifiers cannot really capture other emotions that may overlap between positive or negative class. Hence, the advantage of using fuzzy logic is that one can capture the overlapping values between Yes or No using some linguistic variables. The defined scoring module in the training phase compute the scores of the extracted features (emoticon, hashtag and text) normalized between 0 and 1. The fuzzy logic used these extracted features and their computed scores as input for hate tweet classification. The fuzzy output produces the following Likert scale classification for the determination of a hate tweet sentiment class:

i. Mildly offensive
ii. Moderately offensive
iii. Moderately severe offensive
iv. Severely offensive

The fuzzy extension to tweet classification is made to the following notions and definitions of fuzzy logic:

3.4.2. Fuzzy set

The fuzzy set for the classification of a hate tweet into sentiment classes consist of the features extracted in the semantic analysis of the training phase as the membership set (see Eq. (17)). The members of the defined membership set have different degree of presence between the interval of 0 and 1.

$$A = \{text, emoticon, hashtag\} \tag{17}$$

3.4.3. Linguistic variables

The linguistic variables denotes the degree of membership for the defined membership set A as shown in Eq. (18). It was used to show the degree of classification for a given hate tweet.

$$A(x) = \{mildly, moderately, moderatelysevere, severe\} \tag{18}$$

Table 4
Fuzzification process view.

Linguistic Value	0,if $x =$ a	$\frac{x-a}{c-a}$, if$x \in$ [a, c]	$\frac{b-x}{c-b}$, if $x \in$ [b, c]	0, if x $\geq$ c
Mildly	0,if$x =$ 0.1	$\frac{x-0.1}{0.2}$,if $x \in$ [0.1, 0.3]	$\frac{0.2-x}{0.1}$,if $x \in$ [0.2, 0.3]	0, if $x \geq$ 0.3
Moderately	0,if$x =$ 0.3	$\frac{x-0.3}{0.3}$,if $x \in$ [0.3, 0.6]	$\frac{0.45-x}{0.15}$,if x $\in$[0.45,0.6]	0, if $x \geq$ 0.6
Moderately severe	0,if$x =$ 0.6	$\frac{x-0.6}{0.2}$, if $x \in$ [0.6, 0.8]	$\frac{0.7-x}{0.1}$, if $x \in$ [0.7, 0.8]	0, if $x \geq$ 0.8
Severely	0,if$x =$ 0.8	$\frac{x-0.8}{0.2}$, if $x \in$ [0.8, 1.0]	$\frac{0.9-x}{0.1}$, if $x \in$ [0.9, 1.0]	0, if $x \geq$ 1.0

3.4.4. Fuzzification

The triangular membership function as shown in Eq. (19) was adapted since the membership set consist of three members. Fuzzification was used to change the crisp values to fuzzy values. The fuzzy range of values for the fuzzification process computed by Eq. (19) is shown in Table 3. The fuzzification process view for Eq. (19) is also shown in Table 4.

$$\mu_A(x;[a,b,c]) = \begin{cases} 0, ifx = a \\ \frac{x-a}{c-a}, ifx \in [a,c] \\ \frac{b-x}{c-b}, ifx \in [b,c] \\ 0, ifx \geq c \end{cases} \tag{19}$$

where x represent the x- coordinate of real values and a, b, c represent the y- coordinate between 0 and 1.

3.4.5. Fuzzy rules

A total of 16 rules were defined by the rule of Thumb. Since the linguistic variables used were 4, then we have $2^4 = 16$ rules. The rules were defined by the help of experts in the domain and it is shown in Table 5. The adapted fuzzy logic used the AND function to evaluate its rules by taking the minimum values.

3.4.6. Inference engine

The fuzzy inference engine uses the notion of the fuzzy rules defined on the membership set (text, emoticon, hashtag) for hate tweet sentiment classification. The purpose of these fuzzy rules are to predict the sentiment class for a hate tweet based on the computed semantic scores of Eqs. (8)–(10). The fuzzy inference method used the Root Mean Square (RMS) for reasoning. The RMS formula is given by Eq. (20).

$$\sqrt{\sum R^2} = \sqrt{R_1^2 + R_2^2 + R_3^2 + \ldots + R_n^2} \tag{20}$$

$R_1^2 + R_2^2 + R_3^2 + \ldots + R_n^2$, are values of different rules with the same conclusion in the fuzzy rule base. RMS sum all the resultants of the same firing rules and compute the center of gravity.

3.4.7. Defuzzification

The defuzzification method that was used is Centre of Gravity (CoG) as shown in Eq. (21). The CoG method was adapted because of its

Table 5
Sample rule base for hate tweet classification.

#no	Emoticon	Text	Hashtag	Tweet classification (Conclude)	Non Zero Min no
1	0.25	0.25	0.25	Mildly offensive	0.25
2	0.25	0.5	0.5	Moderately offensive	0.25
3	0.25	0.75	0.75	Moderately severe offensive	0.25
4	0.25	0.9	0.9	Severely offensive	0.25
5	0.5	0.25	0.25	Moderately offensive	0.25
6	0.5	0.5	0.5	Moderately offensive	0.5
7	0.5	0.75	0.75	Severely offensive	0.5
8	0.5	0.9	0.9	Severely offensive	0.5
9	0.75	0.25	0.25	Moderately offensive	0.25
10	0.75	0.5	0.5	Moderately severe offensive	0.5
11	0.75	0.75	0.75	Moderately severe offensive	0.75
12	0.75	0.9	0.9	Severely offensive	0.75
13	0.9	0.25	0.25	Moderately offensive	0.25
14	0.9	0.5	0.5	Moderately severe offensive	0.5
15	0.9	0.75	0.75	Severely offensive	0.75
16	0.9	0.9	0.9	Severely offensive	0.9

Table 6
Hate speech detection.

Tweet Jaccard	Degree of Support	Sample hate speech detected
0.028776978417266189	0.0013703323055841042	@NotoriousBM95: @_WhitePonyJr_ Ariza is a snake and a coward\“ but at least he isn’t a cripple like your hero Roach lmaoo
0.0066666666666666671	0.00033333333333333338	@RTNBA: Drakes new shoes that will be released by Nike/Jordan……. Yes, there’s glitter on the shoes http://t.co/QCtPLxHEXM“ ….dudes a fag
0.038461538461538464	0.0034965034965034965	@ashlingwilde: @ItsNotAdam is bored supposed to be cute, you faggot?\“ Sometimes”
0.0092592592592592587	0.0010288065843621398	@jgabsss: Stacey Dash won &#128166; http://t.co/PDLG46rjOL\“ baddest bitch evaaaa
0.020618556701030927	0.0010309278350515464	Don’t worry about the nigga you see, worry about the nigga you DON’T see… Dat’s da nigga fuckin yo bitch.
0.0089285714285714281	0.00044642857142857141	Hey go look at that video of the man that found the kidnapped girls in Ohio……….what a nigger\“ - #shitmybosssays
0.018633540372670808	0.00098071265119320039	Nah its You @NoMeek_JustMilz: &#128514;&#128514;&#128514;&#128514; yo i thought some1 photoshopped my face on that faggot smmfh…i hate yall
0.036231884057971016	0.0012939958592132505	Our people\“. Now is the time for the Aryan race 2 stand up and say \”no more\“. Before the mongerls turn the world into a ghetto slum. 1488
0.0071942446043165471	0.0002664535038635758	Why people think gay marriage is okay is beyond me. Sorry I don’t want my future son seeing 2 fags walking down the street holding hands an
0.046875	0.0066964285714285711	You ain’t gunna do shit spear chucker
0.016129032258064516	0.0008960573476702509	ayo i even kill handicapped and crippled bitches/look at my scalp real close and you’ll see triple sixes
0.023809523809523808	0.001488095238095238	on my way to fuck your bitch in the name of The Lord\“ - Mr. Race

simplicity and accuracy. Defuzzification is the transformation from fuzzy values to crisp value for better understanding by human. The value from the defuzzification step is meant to take decision on the sentiment class of a given hate tweet.

$$\mathrm{CoG}(Y^{*}) = \frac{\sum \mu \mathrm{y}(X_i)x_i}{\sum \mu \mathrm{y}(X_i)} \tag{21}$$

where $\mu \mathrm{y}(X_i)$ denotes the root mean square for rules having the same conclusion andx_i denotes the mid-points of their respective fuzzy value range.

3.4.8. *User interface*

The second component of the deployment phase is the user interface. It is used to establish interaction between users and the system for further interpretation on the validity of experimental results.

4. Implementation, results and discussion

The designed hate speech classification method was developed using .NET Core framework due to its portability for building applications for all operating systems. Visual Studio 2019 was used as the Integrated Development Environment (IDE) while C# was used as the programming language. Twitter API was used to extract the needed dataset. MATLAB R2012b was used to model and generate the fuzzy rules for sentiment classification. Digital Ocean was used as the cloud service provider for running the program.

Table 6 contain a list of sample tweet Jaccard similarity and BSF with sample hate speech detected. In the training phase, the BirdDog database consist of the labeled dataset as hate tweets while the DecaHose database consist of all tweets including those not relating to hate speech. These two databases where used as input into the combinatorial algorithm to further extract hate speech with Jaccard similarity close to the BirdDog database using the degree of support in BSF. A BSF value equal to or greater than 0.00026 indicates a more likelihood of a tweet been classified as hate tweet and vice versa.

Table 7 depicts sample wordngrams with their frequency of occurrences in a tweet. Feature extraction was performed on the extracted hate tweets dataset called seeders in the training phase. The feature extraction method, extracts the aspect (adjective, verb and adverb) from the dataset. The probabilistic topic spotting measure based on Bayes classifier was also used to infer topics from preprocessed labeled tweets in order to enhance the feature extraction method. Later, the adjective, verb and adverb were categorized into emoticons, texts and hashtags groups. Tri-gram and wordngram models were used to extract the adjective, verb and adverb and separates it. It discards the preceding and successive word occurring with the adjective, verb and adverb in the sentences. For example, the wordngrams of the first tweet in Table 7 will extract “fucking” from the wordngram “you’re fucking” by discarding “you’re” from it. This will be done for all the wordngrams until all the adverbs, verbs or adjectives have been extracted, separated and grouped as either emoticons, texts or hashtags. The purpose of this extraction is to allow for a 4-level scale fuzzy model in the classification phase. In other words, the extracted features and their computed semantic scores serve as input into the fuzzy logic system for a 4-level scale fuzzy classification as depicted in Table 5.

Table 8 depicts the data representation and clustering modules sample results of the clustering phase. The data representation module is responsible for tweet-topic grouping. The probabilistic topic spotting measure based on Bayes classifier and score normalization technique was used to model the likelihood of a tweet fitting to a topic based on the

Table 7
Selected wordngrams and frequencies.

Hate tweet	Wordngram: frequency
@DevilGrimz: @VigxRArts you’re fucking gay, blacklisted hoe“ Holding out for #TehGodClan anyway http://t.co/xUCcwoetmn	“@devilgrimz: @vigxrarts ”:1,“@vigxrarts you’re ”:1,“you’re fucking ”:1,“fucking gay, ”:1,“gay, blacklisted ”:1,“blacklisted hoe\” “:1,”hoe\“ holding ”:1,“holding out ”:1,“out for ”:1,“for #tehgodclan ”:1,“#tehgodclan anyway ”:1,“anyway http://t.co/xuccwoetmn ”:1
@MarkRoundtreeJr: LMFAOOOO I HATE BLACK PEOPLE https://t.co/RNvD2nLCDR“ This is why there’s black people and niggers	“@markroundtreejr: lmfaoooo ”:1,“lmfaoooo i”:1,“ihate ”:1,“hate black ”:1,“black people ”:2,“people https://t.co/rnvd2nlcdr” “:1,“https://t.co/rnvd2nlcdr\” this “:1,”this is “:1,”is why “:1,“why there’s ”:1,“there’s black ”:1,“people and ”:1,“and niggers ”:1
@NotoriousBM95: @_WhitePonyJr_ Ariza is a snake and a coward\“ but at least he isn’t a cripple like your hero Roach lmaoo	“@notoriousbm95: @_whiteponyjr_ ”:1,“@_whiteponyjr_ ariza ”:1,“ariza is ”:1,“is a”:1,“asnake ”:1,“snake and ”:1,“and a”:1,“acoward\” “:1,”coward\“ but ”:1,“but at ”:1,“at least ”:1,“least he ”:1,“he isn’t ”:1,“isn’t a”:1,“acripple ”:1,“cripple like ”:1,“like your ”:1,“your hero ”:1,“hero roach ”:1,“roach lmaoo ”:1

Table 8
Sample detected topics.

Sample hate tweet	Spotting measure	Detected topic
“:”@DevilGrimz: @VigxRArts you’re fucking gay, blacklisted hoe\“ Holding out for #TehGodClan anyway http://t.co/xUCcwoetmn	18.995548822085151	TehGodClan
	17.89693653341704	Stinking
	16.79832424474893	UCFPINKPARTY
	16.79832424474893	ComeAtMeUT
	16.79832424474893	GerrysHalloweenParty
	16.79832424474893	History
	14.233374887287393	Teaparty
	17.89693653341704	Azflooding
	17.89693653341704	BREAKING
	16.79832424474893	Strong
	15.412029883629041	California
	17.491471425308877	DegenerateArtist
	16.392859136640766	Morning
	16.79832424474893	Jerryreed
	15.412029883629041	FireCashman
	16.79832424474893	HOLIDAYSEASONLIVE
	16.105177064188986	HappyColumbusDay
	16.79832424474893	Colonialism
	16.79832424474893	HelloBrookland
	16.105177064188986	HolySpirit
	16.79832424474893	IndigenousPeoplesDay
	14.601099667412711	Redskins
	16.105177064188986	JesusChrist
	14.601099667412711	Faggots
	16.79832424474893	SonOfGod

terms in the tweet. The task is to monitor incoming stream of arriving tweets and automatically group them into the most suitable topic cluster based on their computed Jaccard similarity measure. A tweet is considered belonging to a topic cluster if it’s computed Jaccard similarity measure is greater or equal to the normalized score. Hence, Table 8 represents sample tweets, computed spotting measure and detected topics.

After the collection of labeled hate tweets dataset, feature extraction, topic grouping and clustering, sentiment analysis was used on the tweets and their extracted features. Sentiment analysis is performed using fuzzy logic. The features extracted serves as input to the fuzzy logic system along with the semantic score for each input feature. Words or lexicons relating to hate tweets will have a semantic score close to 1 and lexicons with less hate terms will have a semantic score close to 0.

Table 9 depicts sample results of the classification module in the classification phase. The table shows sample hate tweets and fuzzy classification into one of mildly, moderately, moderately severe or severely offensive linguistic variables. The features extracted from the training phase using the trigrams and wordngrams models enhanced with topics inferred by naïve Bayes classifier were used as the inputs into the developed fuzzy logic in the classification module. A feature list of text, emoji and hashtag groups were obtained from these extracted

Table 9
Hate tweet classification.

Sample hate tweet	Fuzzy value	Sentiment class
@DevilGrimz: @VigxRArts you’re fucking gay, blacklisted hoe“ Holding out for #TehGodClan anyway http://t.co/xUCcwoetmn	0.63	Moderately severe offensive
@MarkRoundtreeJr: LMFAOOOO I HATE BLACK PEOPLE https://t.co/RNvD2nLCDR\“ This is why there’s black people and niggers	0.81	Severely offensive
@NotoriousBM95: @_WhitePonyJr_ Ariza is a snake and a coward\“ but at least he isn’t a cripple like your hero Roach lmaoo	0.60	Moderately severe offensive
@ashlingwilde: @ItsNotAdam is bored supposed to be cute, you faggot?\“ Sometimes	0.53	Moderately offensive
@jgabsss: Stacey Dash won &#128166; http://t.co/PDLG46rjOL\“ baddest bitch evaaaa	0.61	Moderately severe offensive
Don’t worry about the nigga you see, worry about the nigga you DON’T see… Dat’s da nigga fuckin yo bitch.	0.83	Severely offensive
Nah its You @NoMeek_JustMilz: &#128514;&#128514;&#128514;&#128514; yo i thought some1 photoshopped my face on that faggot smmfh…i hate yall	0.37	Mildly offensive
Why people think gay marriage is okay is beyond me. Sorry I don’t want my future son seeing 2 fags walking down the street holding hands an	0.51	Moderately offensive
ayo i even kill handicapped and crippled bitches/look at my scalp real close and you’ll see triple sixes	0.57	Moderately offensive
on my way to fuck your bitch in the name of The Lord\“ - Mr. Race	0.56	Moderately offensive
#AZmonsoon lot of rain, too bad it wasn’t enough to wash away the teabagger racist white trash in the state. #Tcot #teaparty #azflooding	0.60	Moderately severe offensive
#Dutch people who live outside of #NewYorkCity are all white trash.	0.43	Mildly offensive
#JesusChrist was STRAIGHT&gt; That’s why the #faggots killed him. #PERIOD #SonOfGod&gt;	0.42	Mildly offensive
#SlightlyAdjusted RT @CapoToHeaven Alls niggers wanna do is fuck, tweet, and drink pineapple soda all day	0.44	Mildly offensive
#SomethingIGetAlot Are you… asian? black? Hawaiian? gay? retarded? drunk?	0.50	Moderately offensive
#ThingsIWillTeachMyChild how to play deez niggas n bitches dat be snakes	0.43	Mildly offensive
#TrayvonMartin referred to Zimmerman as a “creepy ass cracker”. Racist thug.	0.53	Moderately offensive
#firefighter is a job for white trash	0.41	Mildly offensive

features and their semantic scores computed by a scoring module. The score computation used a hybrid method involving lexicons of ngrams from both corpus-based and dictionary based approaches. Words or lexicons relating to hate tweets will have a semantic score close to 1 and less hate tweets will have a semantic score close to 0. After obtaining the semantic scores for each feature groups, a fuzzification process is performed through rule combination (as illustrated in the scoring module section and Table 5) to obtain the final semantic score output for a tweet. The final semantic score output would produce a Likert scale classification for the determination of hate tweet sentiment class as either mildly, moderately, moderately severe or severely offensive. The developed semantic fuzzy logic for hate speech classification is able to go beyond polarity classification of either positive or negative class common to most sentiment analysis methods.

4.1. Evaluation results

Table 10 show a comparative analysis on sentiment classification of some relevant studies in sentiment analysis on the same computational platform. It can be reported in Table 10 that the developed model for sentiment classification outperforms other models with 91.5% F1-score. Similarly, SVM was found with the least F1-score for sentiment analysis.

Table 11 summarizes the comparative analysis on hate speech detection. Different models were applied on the same extracted dataset and the results of AUC, accuracy, precision, recall, and F1-score were recorded. The models under comparison are defined below:

i. Baseline Naive Bayes is the state-of-the-art Naïve Bayes method.
ii. BERT (Waseem & Hovy, 2016) is a pre-trained transformer model for hate speech detection.
iii. Logistic regression (Davidson et al., 2017) is an automated approach to hate speech detection.
iv. Founta et al. (2019) used deep learning architecture based on RNN (Interleaved) by combining learning with interleaving. The method introduced a way that allows the full network training simultaneously on both text and metadata.
v. RNN (Huang, Xing, Dernoncourt & Paul, 2020) is a deep learning approach including CNN and RNN classifiers by using biGRU for hate speech identification.
vi. LSTM (Corazza et al., 2020) is a deep learning method that include LSTM model for hate speech detection.
vii. CNN-GloVe (Ribeiro & Silva, 2019) is a CNN architecture using pre-trained word embeddings for hate speech detection.
viii. GRU-3-CNN (Wiedemann, Ruppert & Biemann, 2019) is a bi-directional GRU layer with a three parallel CNN layers, each with a different kernel sizes and a filter.
ix. Rule-based clustering is the developed approach for hate speech detection. The rule-based clustering method help in the switching between iterative and incremental clustering techniques for real-time detection of hate speech based on the tweet signature, cluster signature and a time distance (as illustrated in Eq. (12) with running examples). The rule-based clustering system performs iterative clustering i.e., a new tweet is added to one of the predefined topic clusters if the Jaccard similarity measure between an incoming tweet and predefined topic clusters is less than or equal to a normalized score. On the other hand, the rule-based clustering system permit incremental clustering i.e., the new tweet is used to form a new topic cluster if the Jaccard similarity measure between an incoming tweet and predefined topic clusters is greater than a normalized score.

The developed rule-based clustering outperforms both the baseline and the state-of-the-art-methods, as shown in Table 11. It was observed that apart from the good performance of the developed rule-based clustering having AUC, accuracy and F1-score of 0.9645, 0.9453 and 0.9256 respectively, approaches that applied deep learning, transformer methods and tweet-level features attained remarkably good results when compared to the other methods. Looking at results in Table 11, it was observed that most of the transformer and deep learning methods achieved good AUC of 0.7974, 0.9152, 0.9381, 0.9472, 0.6752, 0.8917, good accuracy of 0.7853, 0.8913, 0.9243, 0.9567, 0.6960, 0.8372 and good F1-score of 0.7389, 0.8854, 0.8962, 0.8232, 0.6963 and 0.7736 for

Table 10
Comparative analysis on sentiment classification.

Author	Objective	Model	Features	F1-score
Zhang et al. (2011)	Sentiment Analysis	LMS	Unigrams	87.4%
Jiang et al. (2011)	Polarity classification	SVM	Unigrams	77.6%
Maynard & Funk (2012)	Polarity classification	Graph-basedmethod	Unigrams	83%
Go et al. (2009)	Polarity classification	SVM, NB, MaximumEntropy (Ensemble 1)	Unigrams +bigrams	79.3%
Pak & Paroubek (2010)	Polarity classification	SVM, NB, CRF (Ensemble 2)	N-grams +POS	78.5%
Bifet & Frank (2010)	Polarity classification	MultinomialNaïve Bayes	Unigrams	80.4%
Ribeiro et al. (2018)	Polarity classification	Gradient Boost	GloVe	86.9%
Corazza et al. (2020)	Polarity classification	LSTM	Embedding	82.3%
Current research	**4-level classification**	**Semantic fuzzy logic**	**Trigrams + wordngrams**	**91.5%**

The results are obtained on the same computational platform

Table 11
Comparative analysis on hate speech detection.

Method	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Baseline Naive Bayes	0.7053	0.8652	0.8361	0.8652	0.8460
BERT (Waseem & Hovy, 2016)	0.7974	0.7853	0.7287	0.7775	0.7389
Logistic regression (Davidson et al. 2017)	0.8671	0.8863	0.8513	0.8613	0.8714
Interleaved (Founta et al., 2019)	0.9152	0.8913	0.8852	0.8862	0.8854
RNN (Huang, Xing, Dernoncourt & Paul, 2020)	0.9381	0.9243	0.8961	0.8963	0.8962
LSTM (Corazza et al., 2020)	0.9472	0.9567	0.8231	0.8233	0.8232
CNN-GloVe (Ribeiro & Silva (2019)	0.6752	0.6960	0.7082	0.7124	0.6963
GRU-3-CNN (Wiedemann, Ruppert & Biemann, 2019)	0.8917	0.8372	0.7626	0.7746	0.7736
Rule-based clustering	**0.9645**	**0.9453**	**0.9254**	**0.9174**	**0.9256**

The results are obtained on the same computational platform

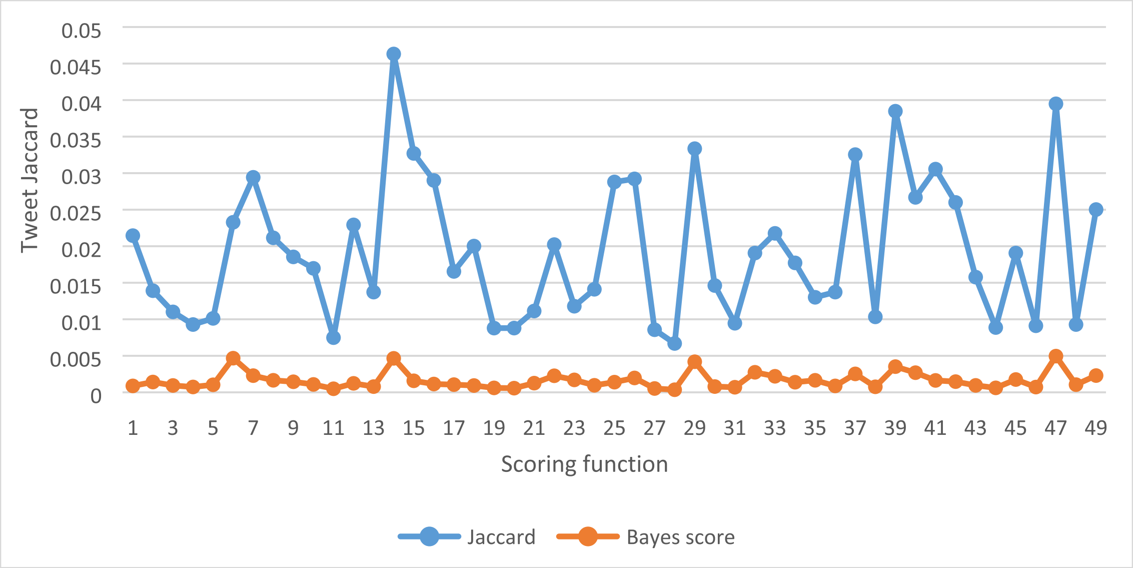

Fig. 6. Tweet-Jaccard Scoring functions.

BERT, Interleaved, RNN, LSTM, CNN-GloVe and GRU-3-CNN respectively. Similarly, Logistic regression performed better with AUC, accuracy and F1-score of 0.8671, 0.8863 and 0.8714 respectively, when compared with the baseline Naïve Bayes, BERT and CNN-GloVe methods. However, the CNN-GloVe method showed the least performances across all metrics compared to the other methods. The results also showed that the developed rule-based clustering produces better results with precision and recall of 0.9254 and 0.9174 respectively, when compared to the other methods.

Fig. 6 present the probabilities of classifying a given tweet as hate tweet using the Jaccard similarity in combinatorial algorithm based on the Bayesian scoring function. The graph shows that a large Jaccard similarity measure will bring about a smaller Bayesian scoring function. The peaks in the graph corresponds to detected hate tweets based on the threshold value. The peaks can also translate to significant changes in hate word frequencies.

Fig. 7 present the comparison of different hate speech detection methods on accuracy, precision, recall and F1-score evaluation metrics. The graph shows that the developed rule-based clustering method performs better across all the metrics with accuracy, precision, recall and F1-score of 0.9453, 0.9254, 0.9174 and 0.9256 respectively, when compared to baseline Naive Bayes, BERT, Logistic regression, Interleaved, RNN, LSTM, CNN-GloVe and GRU-3-CNN with accuracy, precision, recall and F1-score of 0.8652, 0.8361, 0.8652 and 0.8460; 0.7853, 0.7287, 0.7775 and 0.7389; 0.8863, 0.8513, 0.8613 and 0.8714; 0.8913, 0.8852, 0.8862 and 0.8854; 0.9243, 0.8961, 0.8963 and

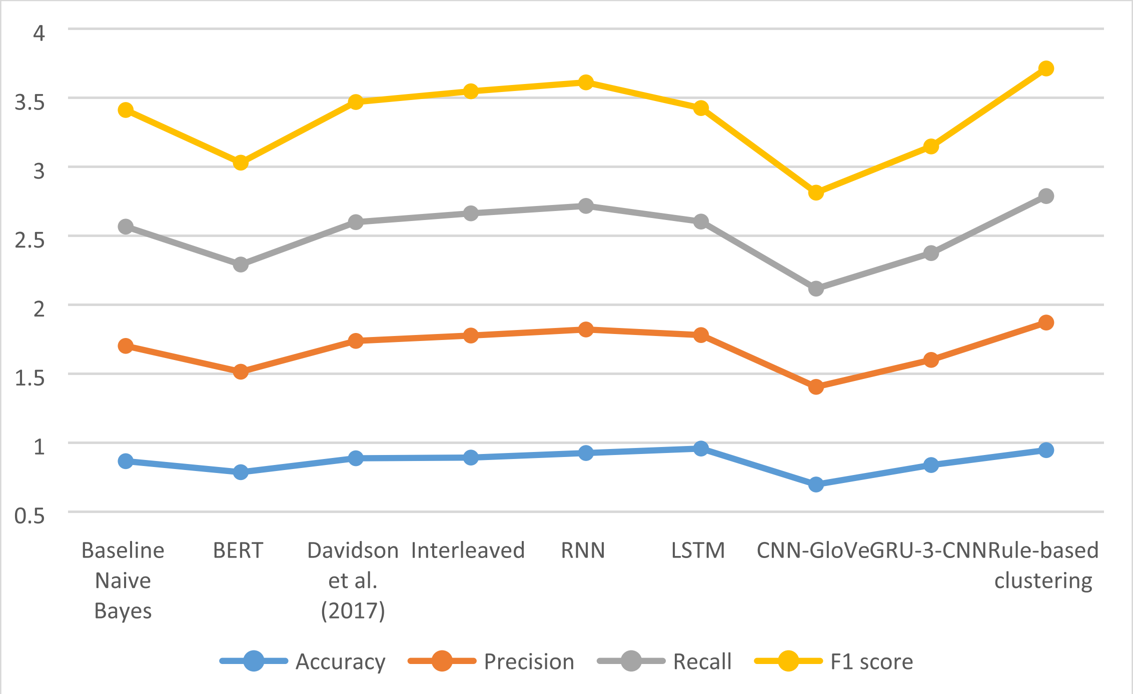

Fig. 7. Hate speech detection models.

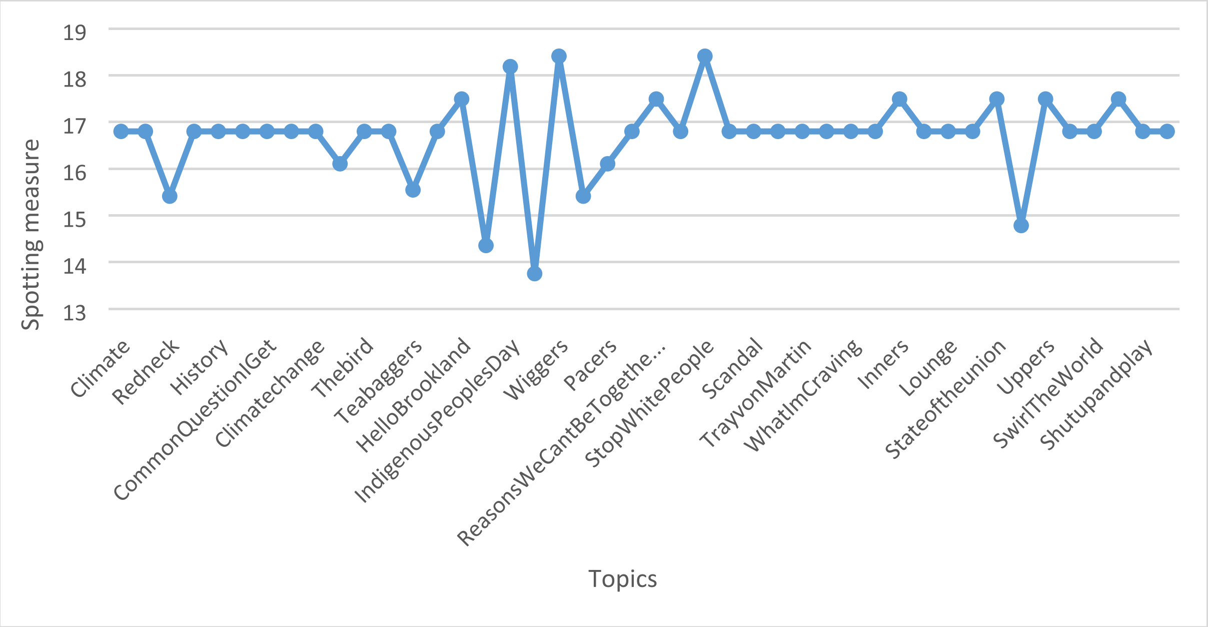

Fig. 8. Tweet-topic detection graph.

0.8962; 0.9567, 0.8231, 0.8233 and 0.8232; 0.6960, 0.7082, 0.7124 and 0.6963; 0.8372, 0.7626, 0.7746 and 0.7736 respectively.

Fig. 8 present the tweet-topic detection graphs. It is apparent that spikes occur on the tweet topics that are bursty and that are hate tweet related. The peaks in the graphs corresponds to detected hate speech, while the peak transitions capture the burst which translate to significant changes in hate word frequencies. The spikes that are flattened represent the proximity of some related hate tweet topics and therefore capture their level of similarity. The figure show the trends over time or categories of hate tweets related topics with few data points. The time interval between peak transitions in the graphs follows an exponential distribution. In other words, the amount of time between occurrences of successive hate topics has an exponential distribution.

Fig. 9 present the comparison of different methods on hate speech classification. The graph shows that the developed semantic fuzzy logic method with F1-score of 91.5% outperforms LMS, SVM, Graph-based, Ensemble 1, Ensemble 2, multinomial naïve Bayes, Gradient Boost and LSTM methods with F1-score of 87.4%, 77.6%, 83%, 79.3%, 78.5%, 80.4%, 86.9% and 82.3% respectively. The high performance of the developed semantic fuzzy logic method can be attributed to the hybrid feature extraction method (trigrams and wordngrams), the use of both text and metadata features and coupled with the use of fuzzy Likert scale

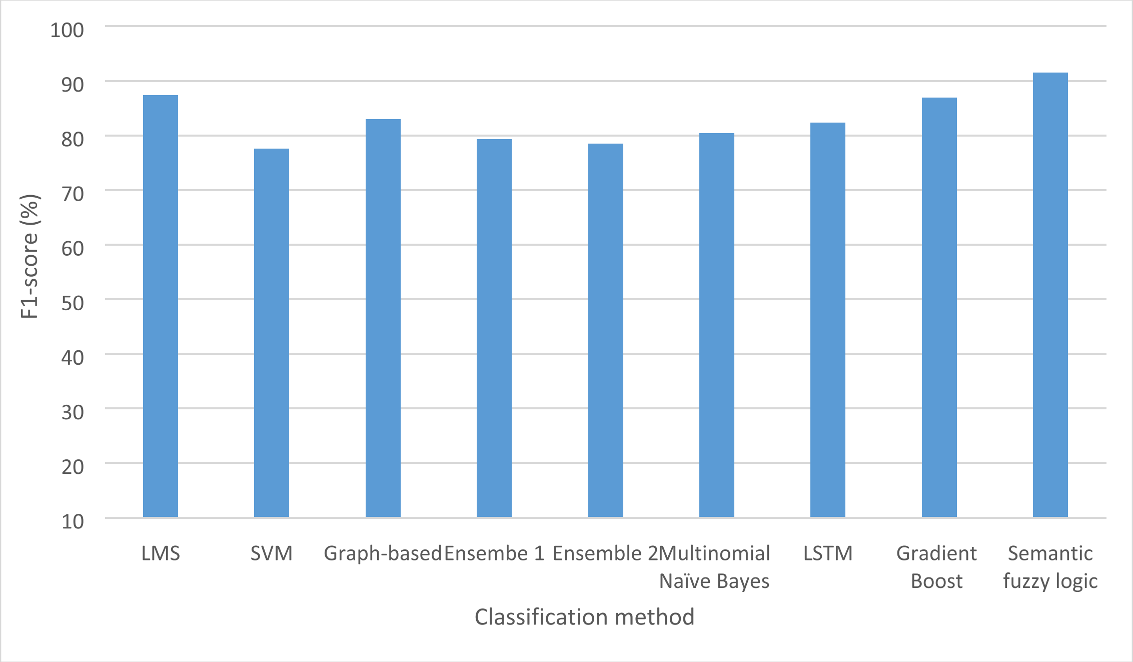

Fig. 9. Comparison of sentiment classification models.

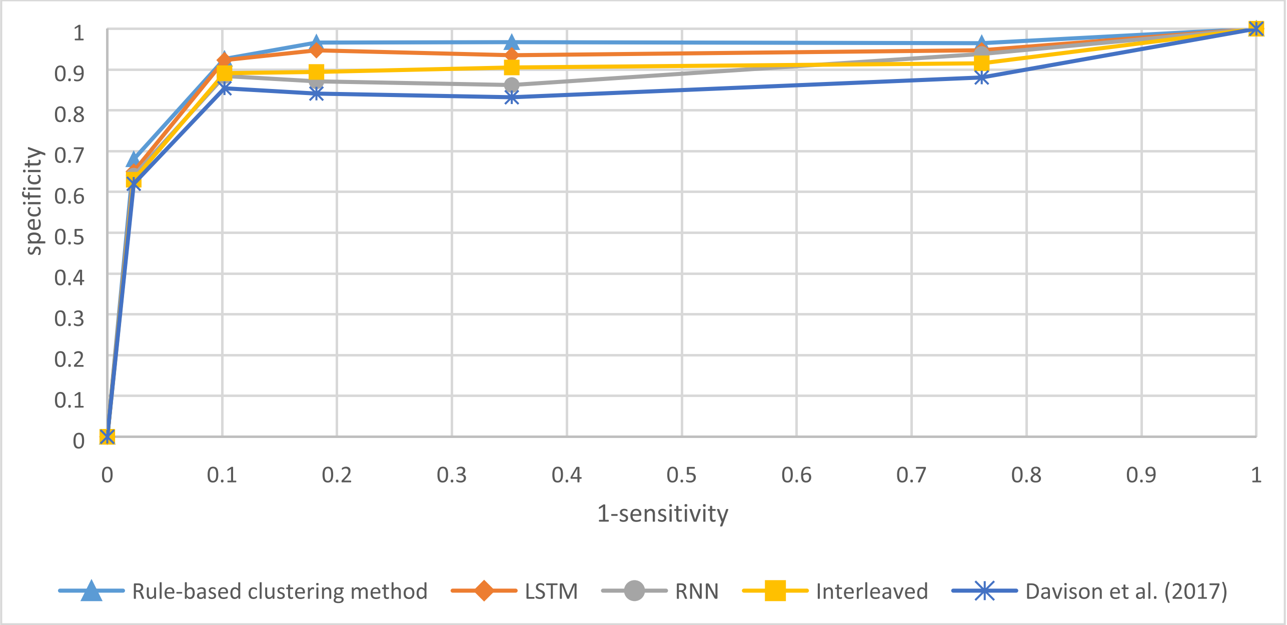

Fig. 10. AUC Comparison.

Table 12
Confusion matrix of the collected dataset.

Label	**Predicted**		
	Hate (%)	Non-hate (%)	Total (%)
Hate (%)	16	7	23
Non-hate (%)	4	73	77
Total (%)	20	80	100

for better sentiment classification.

Fig. 10 present the AUC of different methods on hate speech detection. The AUC is a metric known for its ability to check the quality of detection algorithms. The AUC is a plot of the specificity (positive ratio) and the sensitivity (false positive ratio). The area under the curve measures the overall ability of a method to differentiate between classes of hate tweets. The developed rule-based clustering method indicates a more perfect test having an area of 0.9645 (its AUC nears the upper left corner of the plot), compared to LSTM, RNN, Interleaved and Davison et al. (2017) with an area of 0.9472, 0.9381, 0.9152 and 0.8671 respectively.

The resources needed to fine-tune and reproduce the developed probabilistic clustering model for hate speech classification in twitter are available at: https://github.com/emmini/Twitter-hate-project

4.2. Complexity and error analysis

The simulation results showed that the complexity of the developed probabilistic clustering model for hate speech classification grows quadratically as its input size increases, therefore the developed rule-based clustering method can be said to be of order O(n^2).

As can be observed in the results and discussion section, although the developed hate speech classification model produces very interesting results in terms of F1-score, it is important to analyze the classification errors of the model. In order to understand better the classification error, the test datasets and their resultant confusion matrices from the rule-based clustering method of the developed model is depicted in Table 12. According to Table 12 for the collected dataset, it is obvious that the model can separate hate from non-hate content properly. Only 7% of the total samples belonging to hate class are misclassified as non-hate. Similarly, only 4% of the total samples belonging to non-hate class are misclassified as hate class. It can be concluded that a large majority of the errors come from misclassifying hate class as non-hate class. Hence, the purpose of using fuzzy logic in the classification phase to correctly classify hate speech into a 4-level Likert scale.

5. Conclusion and future work

Hate speech is an expression of intense hatred and thus the need to develop methods to control such negative expression. In this study, a probabilistic clustering model for hate speech classification in twitter was developed. The developed model is divided into four phases namely metadata representation, training, clustering and classification. The use of an automatic topic spotting measure based on naïve Bayes model to improve features representation was introduced. The clustering task include the design of a rule-based system for clustering real-time tweet into topic clusters based on a modified Jaccard similarity measure to reduce misclassification. In order to further reduce misclassification in hate class, a 4-level scale fuzzy logic was used for hate speech classification. From the evaluation results, it was observed that the developed model performs better across all the metrics when compared to the other hate speech detection models. Similarly, the developed model also performed well in hate speech classification with interesting results compared to the other models. The results also showed that the developed model is good for topic detection and categorization.

The advantage of the developed probabilistic clustering model for hate speech classification is based on its robustness to data imbalance, imprecision, threshold setting and fragmentation issues. On the other hand, the limitation of the developed probabilistic clustering model for hate speech classification is that it is very sensitive to the size of the training data. In order words, the complexity of the model increases with increase in the size of the dataset.

In the future, research can be conducted to implement a visualization screen for the sentiment classification phase. Future work will also deploy supervised learning to predict the sentiment of new tweet from dataset with sentiment labels. The fuzzy logic linguistic variables will be increased to accommodate more sentiment classes for better classification accuracy. Machine learning algorithms can also be deployed for multilingual hate speech detection in Nigeria Twitter.

Author contribution statement

F. E. Ayo: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

O. Folorunso: Performed the experiments; Wrote the paper.

F. T. Ibharalu, I. A. Osinuga, A. Abayomi-Alli: Contributed data; Analyzed and interpreted the data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix I:. Metadata extraction algorithm

```
Input: $tw_1, tw_2, \cdots, tw_n$ // source metadata
Output: structured text file
begin
  $C_k \leftarrow tw_1, tw_2, \cdots, tw_n$ // get source metadata
  for tw in $C_k$ do
  unstructuredText← typeOfInitial data // print the structure of original data
  structuredText = json.loads(unstructuredText)
  parseText← D(structuredText)
  end for
  for tw in parseText do
    return required features in a structured text file
  end for
end
```

Appendix II. Scoring function

```
Input: unknown tweet: $t_i$; hate tweet profile: $p_i$
Output: $i \rightarrow$ hate tweets; $f \rightarrow$semantic features
Combinatorial pattern matching ($t_i$, $p_i$)
begin
k← normalized threshold value
n ← length of hate tweet pattern $p_i$
m ← length of unknown tweet $t_i$
foreach tweet tw in T do //filtering method
    $s_w \leftarrow$ STOPWORD_TWEET($t$)
    $u_{rl} \leftarrow$ URL_TWEET($t$)
    if $ns \leftarrow max(s_w; u_{rl}; \cdots)$ //maximize the noise level
    if $ns$ = noise then
        $ns \leftarrow$ Noise_Tweet(t)
        $T_{\neg ns} \leftarrow tw - ns$ //obtain non-noise hate speech tweets
      end if
  end if
end foreach
for $i \leftarrow 1$ to $m - n + 1$
    $J(t,p) = (t \cap p)/(tUp)$ //Jaccard index
    if$t_i = p$
        $S(t,p) = n_t(p)$ //degree of support
        foreach tweet tw in T, obtain trigrams and wordngrams
            extract features emoji, hashtag, and text
            f (emoji, hashtag, and text) ←compute scores using (8), (9) and (10)
            if $S(t,p) >= k$ then
                T←output i, f //obtain hate tweets and features
            end if
        end foreach
    end if
end for
end begin
```

Appendix III. Data representation

```
Input: Tweet: $S_i$, Topic: $T_j$ and features: $f$
Output: Topic clusters: C
Probabilistic Topic Spotting Measure $(S, T)$
begin
$T_1 \leftarrow S_1$; $S, T \in T_{\neg ns}$
$C \leftarrow 1$
for $i \leftarrow 2 to N$
  for $j \leftarrow 1 to C$
    compute $M(S_i, T_j)$ using Eq. (11)
```

(continued on next page)

(*continued*)

```
        $N_{select}$←compute normalized scores using Eq. (13)
    $k$← $ARGMAX_J$ (M($S_i$, $T_J$)) //cluster with max similarity
        end for
end for
if (M($S_i$, $T_k$))<= $N_{select}$
    Assign $S_i$ to topic $T_k$
else
 $C$←$C$ + 1
    Create new cluster $T_C$ from $S_i$
    Re-estimate all of $T_i$ for $i$← 1 to C17.end if
end begin
```

Appendix IV. Rule-based clustering

```
Input: A continuous stream of tweets $tw_1, \cdots, tw_n, \cdots$
       normalized similarity threshold $\varepsilon$
       Topic clusters, $C_1, \cdots, C_k$
Output: The set of currently active clusters C
begin
$tw_1$ ← GenerateTweetObject($tw_1$)
i = 1
while (not end of stream)
    i ← i + 1;
    Receive the next tweet $tw_i$
    $tw_i$ ← GenerateTweetObject($tw_i$)
    foreach cluster $C_j \in C = \{C_1, \cdots, C_k\}$
        if isActive($C_j$) then
            sim($tw_i$, $C_j$) ← ComputeSimilarity using eq. (12)
        else
            C ← C − $C_j$
    end if
        end foreach
    c =argmax$_{j\in\{1,\cdots,k\}\mathrm{sim}(tw_i,C_j)}$//return cluster with max score
    compute $\varepsilon(C_j)$ using eq. (13)
    if sim($tw_i$, $C_j$) <= $\varepsilon$ then
        updateCentroid($C_c$ , $tw_i$)//iterative; add tweet to topic cluster
    elseif sim($tw_i$, $C_j$) > $\varepsilon$
    $C_1$ ← CreateCluster($tw_1$)//incremental; form new topic cluster
        C ← C∪ $C_1$;
    end elseif
    end if
end while
end begin
```

Appendix V. Sentiment classification

```
Input: T← Tweets, $f$← extracted features,$S_f$←semantic score
Output: $C_k$←Tweet sentiment class
        begin
        foreach tweet t ∈ T
            $f$←Extract features F from preprocessed dataset using tf-idf and naïve Bayes
            foreach $f_i$to$f$-length do
        $S_f$←ComputeScores using Eqs. (8)–(10)
        //Apply fuzzy logic on features and scores
        for all $\langle f, S_f\rangle \in f$ do
            if $\left(0.1 \le S_f < 0.3\right)$ then
                $C_k$← mildly offensive
              elseif $\left(0.3 \le S_f < 0.6\right)$ then
                  $C_k$← moderately offensive
              elseif $\left(0.6 \le S_f < 0.8\right)$ then
                  $C_k$← moderately severe ofensive
              elseif $\left(0.8 \le S_f \ge 1.0\right)$ then
                  $C_k$← severely offensive
              end if
          end for all
          end foreach
    end foreach
    return $C_k$
    end begin
```

References

Araújo, M., Pereira, A., & Benevenuto, F. (2020). A comparative study of machine translation for multilingual sentence-level sentiment analysis. *Information Sciences, 512*, 1078–1102.

Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence, 31*(1), 132–164.

Badlani, R., Asnani, N., & Rai, M. (2019). Disambiguating sentiment: An ensemble of humour, sarcasm, and hate speech features for sentiment classification. *W-NUT, 2019*, 337.

Bellan, P., & Strapparava, C. (2018). Detecting Inappropriate Comments to News. In *International Conference of the Italian Association for Artificial Intelligence* (pp. 403–414). Cham: Springer.

Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information FusionElsevier BV. Netherlands, 28*, 45–59.

Bifet, A., & Frank, E. (2010). In *Sentiment knowledge discovery in twitter streaming data* (pp. 1–15). Berlin, Heidelberg: Springer.

Bisht, A., Singh, A., Bhadauria, H. S., & Virmani, J. (2020). Detection of hate speech and offensive language in twitter data using LSTM model. In *Recent trends in image and signal processing in computer vision* (pp. 243–264). Singapore: Springer.

Bonini, T., Caliandro, A., & Massarelli, A. (2016). Understanding the value of networked publics in radio: Employing digital methods and social network analysis to understand the Twitter publics of two Italian national radio stations. *Information, Communication & Society, Taylor & Francis, 19*(1), 40–58.

Bosco, C., Felice, D. O., Poletto, F., Sanguinetti, M., & Maurizio, T. (2018). Overview of the EVALITA 2018 hate speech detection task. In EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (Vol. 2263, pp. 1-9). CEUR.

Brzozowski, M. J., & Romero, D. M. (2011). Who should I follow? Recommending people in directed social networks. In *Proceedings of the fifth international AAAI conference on weblogs and social* (pp. 458–461).

Burnap, P., & Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet, 7*(2), 223–242.

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In Proceedings of the 20th international conference on World wide web, WWW '11, ACM, New York, NY, pp. 675–684.

Chang, Y. C., Ku, C. H., & Chen, C. H. (2019). Social media analytics: Extracting and visualizing Hilton hotel ratings and reviews from TripAdvisor. *International Journal of Information Management, 48*, 263–279.

Chau, M., & Xu, J. (2007). Mining communities and their relationships in blogs: A study of online hate groups. *International Journal of Human-Computer Studies, 65*(1), 57–70.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. ACL.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In NIPS 2014 Workshop on Deep Learning, arXiv preprint arXiv:1412.3555.

Cimino, A., De Mattei, L., & Dell'Orletta, F. (2018). Multi-task learning in deep neural networks at evalita 2018. Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), Turin, Italy. CEUR.org, pp.86-95.

Corazza, M., Menini, S., Arslan, P., Sprugnoli, R., Cabrio, E., Tonelli, S., & Villata, S. (2018). Inriafbk at germeval 2018: Identifying offensive tweets using recurrent neural networks.

Corazza, M., Menini, S., Cabrio, E., Tonelli, S., & Villata, S. (2020). A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology (TOIT), 20*(2), 1–22.

Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management, Elsevier Ltd., United Kingdom, 57*(2), 102034. https://doi.org/10.1016/j.ipm.2019.04.002

Dang, Y., Zhang, Y., & Chen, H. (2009). A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intelligent Systems, 25*(4), 46–53.

Daniel, M., Neves, R. F., & Horta, N. (2017). Company event popularity for financial markets using Twitter and sentiment analysis. *Expert Systems with Applications, Elsevier, 71*, 111–124.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Eleventh International AAAI Conference on Web and Social Media*, 512–515.

de Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. arXiv preprint arXiv:1809.04444.

Earl, J., & Garrett, R. K. (2016). The new information frontier: toward a more nuanced view of social movement communication. Social Movement Studies, Taylor & Francis, pp.1-15.

Florio, K., Basile, V., Polignano, M., Basile, P., & Patti, V. (2020). Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences, 10*(12), 4180.

Folorunso, O., Ayo, F. E., & Babalola, Y. E. (2016). Ca-NIDS: A network intrusion detection system using combinatorial algorithm approach. *Journal of Information Privacy and Security, 12*(4), 181–196.

Fortuna, P., da Silva, J. R., Wanner, L., & Nunes, S. (2019). A hierarchically-labeled Portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 94–104).

Founta, A. M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., & Leontiadis, I. (2019). A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science* (pp. 105–114).

Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., et al. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. *Twelfth International AAAI Conference on Web and Social Media*.

Fullér, R., Hassanein, H., & Ali, A. N. (1996). Neural fuzzy systems: towards IMT-advanced networks. Åbo: Åboakademi xxvii, 275 p. ISBN 95-165-0624-0.

Gitari, N. D., Zhang, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering, Science and Engineering Research Support Society, South Korea, 10*(4), 215–230.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12), 2009.

Greenwood, M. A., Bakir, M. E., Gorrell, G., Song, X., Roberts, I., & Bontcheva, K. (2019). Online Abuse of UK MPs from 2015 to 2019. pp. 1–18.

Grover, P., Kar, A. K., Dwivedi, Y. K., & Janssen, M. (2019). Polarization and acculturation in US Election 2016 outcomes–Can twitter analytics predict changes in voting preferences. *Technological Forecasting and Social Change, Elsevier BV, Netherlands, 145*, 438–460.

Howells, K., & Ertugan, A. (2017). Applying fuzzy logic for sentiment analysis of social media network data in marketing. *Procedia Computer science, 120*, 664–670.

Huang, X., Xing, L., Dernoncourt, F., & Paul, M. J. (2020). Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. arXiv preprint arXiv:2002.10361.

Hurlock, J., & Wilson, M. L. (2011). Searching Twitter: Separating the Tweet from the Chaff. In International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, pp. 161–168.

Ibrohim, M. O., & Budi, I. (2019). Multi-label hate speech and abusive language detection in Indonesian twitter. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 46–57).

i-Orts, Ò. G. (2019). Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 Task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 460–463).

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology, 60*(11), 2169–2188.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1, 151–60. Stroudsburg, PA: Association for Computational Linguistics.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons, Elsevier, 53*(1), 59–68.

Khreich, W., Granger, E., Sabourin, R., & Miri., A.. (2009). Combining hidden Markov models for anomaly detection. In *International Conference on Communications (ICC)* (pp. 1–6). Germany: Dresden.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? In Proceedings of the 19th international conference on World wide web, New York, NY, ACM, pp. 591–600.

Kwok, I., & Wang, Y. (2013). Locate the hate: Detecting tweets against blacks. *Twenty-seventh AAAI Conference on Artificial Intelligence*, 1621–1622.

Lee, E.-J., Lee, H.-Y., & Choi, S. (2020). Is the message the medium? How politicians' Twitter blunders affect perceived authenticity of Twitter communication. *Computers in Human Behavior, Elsevier Ltd, United Kingdom, 104*, 106188. https://doi.org/10.1016/j.chb.2019.106188

Lee, K., Eoff, B. D., & Caverlee, J. (2011). Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In International AAAI Conference on Weblogs and Social Media, Barcelona, Spain.

Liu, K. L., Li, W., & Guo, M. (2012). Emoticon smoothed language models for Twitter sentiment analysis. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Liu, P., Guberman, J., Hemphill, L., & Culotta, A. (2018). Forecasting the presence and intensity of hostility on Instagram using linguistic and social features. *Twelfth International AAAI Conference on Web and Social Media*.

Lu, X. (2018). Online communication behavior at the onset of a catastrophe: An exploratory study of the 2008 Wenchan earthquake in China. *Natural Hazards, Netherlands, Springer, 91*(2), 785–802.

Mainka, A., Hartmann, S., Stock, W. G., & Peters, I. (2014). Government and social media: A case study of 31 informational world cities. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on IEEE* (pp. 1715–1724).

Maynard, D., & Funk, A. (2012). Automatic detection of political opinions in tweets. In R. Garć ıa-Castro, D. Fensel, and Antoniou, G. (eds.), The Semantic Web: ESWC 2011 Workshops, Lecture Notes in Computer Science, 7117, 88–99. Berlin/Heidelberg: Springer.

Medina, R. Z., & Diaz, J. C. L. (2016). Social Media Use in Crisis Communication Management: An Opportunity for Local Communities? Social Media and Local Governments. Springer International Publishing, pp. 321–335.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography, Oxford University Press, 3*(4), 235–244.

Mulki, H., Haddad, H., Ali, C. B., & Alshabani, H. (2019). L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 111–118).

Nejad, M. Y., Delghandi, M. S., Bali, A. O., & Hosseinzadeh, M. (2020). Using Twitter to raise the profile of childhood cancer awareness month. Network Modeling Analysis in Health Informatics and Bioinformatics, 9(3), 1–5. Springer Nature, United States.

Nip, J. Y., & Fu, K. W. (2016). Networked framing between source posts and their reposts: an analysis of public opinion on China's microblogs. Information, Communication & Society, 19(8), 1127–1149.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of Jaccard coefficient for keywords similarity. *Proceedings of the International Multiconference of Engineers and Computer Scientists, 1*(6), 380–384.

Nockleby, J. T. (2000). Hate speech. In Encyclopedia of the American Constitution (2nd ed., edited by Leonard W. Levy, Kenneth L. Karst et al.). New York: Macmillan, 3, 1277–1279.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In N. C. C. Chair, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, Sentiment analysis in Twitter 27 S, Piperidis, M. Rosner, and D. Tapias (eds.), In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta; ELRA, European Language Resources Association. pp. 19–21.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, Hanover, MA, USA, 2(1–2), 1-135.

Park, J. H., Shin, J., & Fung, P. (2018). Reducing gender bias in abusive language detection. In Proceedings of the 2018 Conference on EMNLP, pp. 2799–2804.

Paschalides, D., Stephanidis, D., Andreou, A., Orphanou, K., Pallis, G., Dikaiakos, M. D., et al. (2020). MANDOLA: A big-data processing and visualization platform for monitoring and detecting online hate speech. *ACM Transactions on Internet Technology (TOIT), 20*(2), 1–21.

Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Detecting offensive language in tweets using deep learning. arXiv preprint arXiv:1801.04433. pp. 1–17.

Polignano, M., Basile, P., de Gemmis, M., & Semeraro, G. (2019). Hate Speech Detection through AlBERTo Italian Language Understanding Model. In 3rd Workshop on Natural Language for Artificial Intelligence (NL4AI) at the 18th International Conference of the Italian Association for Artificial Intelligence, NL4AI@ AI* IA. Rende, Italy, pp. 1–13.

Ptaszynski, M., Pieciukiewicz, A., & Dybała, P. (2019). Results of the PolEval 2019 Shared Task 6: First dataset and open shared task for automatic cyberbullying detection in Polish Twitter. Proceedings of the PolEval 2019 Workshop, 89p.

Ribeiro, A., & Silva, N. (2019). INF-HatEval at SemEval-2019 Task 5: Convolutional neural networks for hate speech detection against women and immigrants on Twitter. In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 420-425).

Ribeiro, M. H., Calais, P. H., Santos, Y. A., Almeida, V. A., & Meira, W., Jr (2018). Characterizing and detecting hateful users on twitter. *Twelfth International AAAI Conference on Web and Social Media*, 676–679.

Sakaki, T., Okazaki, M., & Matsuo, Y. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web, WWW '10, New York, NY: ACM, pp. 851–860.

Sanguinetti, M., Poletto, F., Bosco, C., Patti, V., & Stranisci, M. (2018). An italian twitter corpus of hate speech against immigrants. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., & Sperling, J. (2009). Twitterstand: news in tweets. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems, ACM* (pp. 42–51).

Schnitzler, K., Davies, N., Ross, F., & Harris, R. (2016). Using Twitter™ to drive research impact: A discussion of strategies, opportunities and challenges. *International Journal of Nursing Studies, 59*, 15–26.

Schwartz, R., Imai, T., Kubala, F., Nguyen, L., & Makhoul, J. (1997). A maximum likelihood model for topic classification of broadcast news. In Proc. Fifth European Conference on Speech Communication and Technology, Rhodes, Greece, 3, pp. 1455–1458.

Serra, J., Leontiadis, I., Spathis, D., Stringhini, G., Blackburn, J., & Vakali, A. (2017). Class-based prediction errors to detect hate speech with out-of-vocabulary words. In *Proceedings of the First Workshop on Abusive Language Online* (pp. 36–40).

Setyadi, N. A., Nasrun, M., & Setianingsih, C. (2018). Text analysis for hate speech detection using backpropagation neural network. In 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC) (pp. 159–165). IEEE.

Siddiqua, U. A., Chy, A. N., & Aono, M. (2019). Kdehateval at semeval-2019 task 5: A neural network model for detecting hate speech in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 365–370).

Salton, G., & Yu, C. T. (1973). On the construction of effective vocabularies for information retrieval. *ACM Sigplan Notices, 10*(1), 48–60.

Harris, Z. S. (1954). Distributional structure. *Word, 10*(2-3), 146–162.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *KDD Workshop on Text Mining, Boston, MA, 400*(1), 525–526.

Taberner, R. (2016). e-Dermatology: Social networks and other web based tools. *Actas Dermo-Sifiliográficas (English Edition), Elsevier, 107*(2), 98–106.

Taboada, M., Anthony, C., & Voll, K. D. (2006). Creating semantic orientation dictionaries. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)* (pp. 427–432).

Tian, D., Gledson, A., Antoniades, A., Aristodimou, A., Dimitrios, N., Sahay, R., & Keane, J. (2013). A Bayesian association rule mining algorithm. In Systems, Man, and Cybernetics (SMC): IEEE International Conference, pp. 3258–3264.

Wang, L. X. (1994). Adaptive fuzzy systems and control. Design and stability analysis. Englewood Cliffs, N.J: Prentice Hall, 1994, xxvii, 275 p. ISBN 978-013-1471-092.

Wang, X., Gerber, M. S., & Brown, D. E. (2012). Automatic crime prediction using events extracted from twitter posts. In *International conference on social computing, behavioral-cultural modeling, and prediction, SBP'12* (pp. 231–238). Berlin, Heidelberg: Springer-Verlag.

Warner, W., & Hirschberg, J. (2012). Detecting hate speech on the World Wide Web. In Proceedings of the second workshop on language in social media. Association for Computational Linguistics. pp. 19–26.

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88–93).

Westerman, D., Spence, P. R., & Van Der Heide, B. (2012). A social network as information: The effect of system generated reports of connectedness on credibility on Twitter. *Computers in Human Behavior, 28*(1), 199–206.

Wiedemann, G., Ruppert, E., & Biemann, C. (2019). UHH-LT at SemEval-2019 task 6: Supervised vs. unsupervised transfer learning for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 782–787).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language.

Winter, K., & Kern, R. (2019). Know-center at SemEval-2019 Task 5: Multilingual hate speech detection on Twitter using CNNs. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 431–435).

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 1391–1399).

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-III. *Information sciences, 9*(1), 43–80.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. 2011. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Technical Report HPL-2011-89.