Article

JIS

Journal of Information Science
2021, Vol. 47(4) 483–501
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0165551520917651
journals.sagepub.com/home/jis
SAGE

Intelligent detection of hate speech in Arabic social network: A machine learning approach

Ibrahim Aljarah
The University of Jordan, Jordan

Maria Habib
The University of Jordan, Jordan

Neveen Hijazi
The University of Jordan, Jordan

Hossam Faris
The University of Jordan, Jordan

Raneem Qaddoura
Philadelphia University, Jordan

Bassam Hammo
The University of Jordan, Jordan

Mohammad Abushariah
The University of Jordan, Jordan

Mohammad Alfawareh
The University of Jordan, Jordan

Abstract

Nowadays, cyber hate speech is increasingly growing, which forms a serious problem worldwide by threatening the cohesion of civil societies. Hate speech relates to using expressions or phrases that are violent, offensive or insulting for a person or a minority of people. In particular, in the Arab region, the number of Arab social media users is growing rapidly, which is accompanied with high increasing rate of cyber hate speech. This drew our attention to aspire healthy online environments that are free of hatred and discrimination. Therefore, this article aims to detect cyber hate speech based on Arabic context over Twitter platform, by applying Natural Language Processing (NLP) techniques, and machine learning methods. The article considers a set of tweets related to racism, journalism, sports orientation, terrorism and Islam. Several types of features and emotions are extracted and arranged in 15 different combinations of data. The processed dataset is experimented using Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT) and Random Forest (RF), in which RF with the feature set of Term Frequency-Inverse Document Frequency (TF-IDF) and profile-related features achieves the best results. Furthermore, a feature importance analysis is conducted based on RF classifier in order to quantify the predictive ability of features in regard to the hate class.

Keywords

Hate speech; machine learning; text vectorization; Twitter

Corresponding author:
Hossam Faris, The University of Jordan, Queen Rania Str., Amman, 19328, Jordan.
Email: Hossam.faris@ju.edu.jo

1. Introduction

As we are living in the age of big data, social media and social networking sites (SNSs) are growing at a surpassing rate. SNSs are digital environments where users can communicate, interact and share information. One of the most popular, fast-spreading and micro-blogging services of SNS is Twitter. Apparently, SNSs result in a massive amount of user-generated data that forms a rich resource for conducting research and building a valuable knowledge. However, some users exploit the spread of online social networks for propagating extremist and discrimination ideas which lead to the dissemination of hate speech or hate crime.

Social media mining considers the use of data mining and text mining techniques, alongside with social networking analytic and information retrieval, to extract informative patterns of data that reveal the mystery underneath the diffusion of information, opinions, or sentiments. Interpreting the sentiment orientation of public is very crucial for decision makers in institutions and organisations. Sentiment analysis (SA) is also known as opinion mining which refers to analysing people's opinions, sentiments, emotions and attitudes towards an object, person, service or any other entity [1]. Basically, SA includes analysing a set of textual documents to derive a group of descriptive features that best describe the sentiment of an entity. Mainly in SA, an opinion or sentiment is characterised by determining key elements, which are the object, the aspect, the sentiment orientation, the opinion holder, and the time. The object is, for example, a person, organisation or service, while the aspect represents the object's attribute which is of interest for the opinion holder at certain time. Often, the sentiment is represented as either positive, negative or neutral [2]. Basically, SA is usually performed at different levels of granularity: as document-level, sentence-level, and aspect-level. In document-level SA, each piece of text is treated as an element that holds an opinion towards single object, whereas the sentence-level SA aims at extracting the opinion orientation for smaller piece of text which is more challenging; since the sentiment of words depends strongly on the context of writing, while the Aspect-level SA concerns on identifying the key elements of an opinion with more emphasis on the aspect extraction and aspect sentiment classification [2].

Primarily, SA mechanisms are categorised into supervised approaches, unsupervised approaches, and a hybrid of both. Unsupervised methods depend on lexicon-based methods which use dictionary-based or corpus-based approaches to unveil the semantic orientation of texts by having many sentiment phrases, whereas the supervised approaches depend mainly on using data mining tools and Natural Language Processing (NLP) techniques in order to train a learning algorithm on a set of labelled data [2]. One of the ongoing applications of SA with a surging interest is the detection of hate speech throughout online social networks. Hate speech is the use of such offensive violent expressions with the objective for spreading hatred, discrimination, bullying or intimidation for a person or minority, on the grounds of race, sex, religion or disability. The European Court of Human Rights (ECHR) defines hate speech as any expressions that disseminate, encourage or incite hatred based on race or xenophobia, as well as any form of intolerance towards immigrants or minorities [3].

The evolved SNSs present a powerful environment for freely communicating ideas by posting, tweeting and retweeting media data. However, it is very essential and critical for safeguarding the right for speaking, expressing opinions and emotions, and at the same time avoiding the use of awful, antagonistic, disrespectful phrases towards others. Twitter and Facebook started combating online hate speech by defining policies that restrict the usage of violent and dehumanising speeches [4,5]. Furthermore, several Arab countries have defined different policies and laws for fighting cybercrimes in general and hate speech in particular. For example, the Jordanian cybercrime law [6] defines hate speech as 'any speech, writing, or action that intended to cause sectarian or racial strife, or call for violence and incitement to conflict between sects and various elements of the nation' [7].

In essence, the Arabic language is the fourth used language on the web and is ranked as the sixth used one on Twitter [8]. Arabic language is the official spoken language of two billion Muslims around the world and is the original language of Islam's holy book. Arabic language is a morphologically rich and complex language that exposes the SA processes and the NLP techniques to various obstacles [9]. The Arabic language can be written in different forms: the Modern Standard Arabic (MSA) which is the official language, the Dialectal Arabic (DA), and the Classical Arabic which is the language of Islam's holy book [10]. Although the Arabic language has a relatively wide-spread usage worldwide, it has several characteristics which made the process of SA more challenging than other well-studied languages, such as the English language [10]. Basically, various characteristics of the Arabic language made the process of SA a challenging task. For instance, the use of colloquial Arabic is very challenging because Arabic countries have different dialects yet sometimes the dialects are different across the cities of the same country. Besides the high variation of dialects, DA has many misspellings which differs morphologically and phonologically when compared with MSA making the NLP process (such as the use of part of speech taggers) less effective. In addition, the Arabic language has more complex orthography and morphosyntactic rules than other languages, and rich morphological analysis forms. This results in a lack of thorough Arabic sentiment lexicon resources and tools [9].

Although Arabic SA is an actively growing research direction, it is insufficiently studied throughout literature and it still has several challenges and open research areas. This article investigates the efficiency of machine learning-based approaches for detecting cyber hate speech on Twitter. The authors collected the data from Twitter, preprocessed and annotated it into two classes: 'Positive' and 'Negative', in which the positive represents the hate speech and the negative represents the non-hate speech. Moreover, the article interprets the effects of different text vectorizations techniques which include the Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF) and the bag-of-words (BoW). It also extracts profile-related features such as 'retweet-count', 'favourite-count' and other features. Nonetheless, it converts the emoticons into features. All combinations of the aforementioned features are utilised for training Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT) and Random Forest (RF). The combinations of features include the word-based features (TF, TF-IDF, BoW), profile-related features and the emotion features. For instance, the combination of words features (TF) and emotion feature are used and the profile features are excluded, or the profile and emotions features are used while excluding the words features, and so on. In consequence, the results of the conducted experiments show that the combination of TF-IDF and profile-related features with RF classifier is superior among all combinations, and in comparison with other classifiers in terms of accuracy and geometric mean. Furthermore, a feature importance analysis is conducted based on RF classifier which depends mainly on Gini index and impurity of the dataset. By examining the average decrease of impurity of features, the racism, emigrant and God word are recognised as the most informative features for predicting the target class.

The main contributions of this research article are as follows:

- Creating a dataset designated for capturing hatred expressions on social media sites and in particular Twitter.
- Conducting several text vectorization techniques and machine learning algorithms on the dataset to measure its performance and capability in detecting cyber hate speech.
- Perform feature importance analysis on the collected dataset for determining the most significant informative features.

The rest of the article is organised as follows: Section 2 represents the literature review of SA and cyber hate speech detection. Section 3 includes the background of NLP techniques and machine learning algorithms. Section 4 shows the designed methodology. Section 5 exhibits the results of the conducted experiments. Finally, Section 6 is an overall summary.

2. Literature review

Nowadays, social media sites represent a prominent place to express opinions and feelings towards people, objects or services. Since morphological languages have been sparsely studied in recent years, this review emphasises on cyber hate speech and SA of Arabic language as rich and complex language, as well as it encompasses a review of English and other foreign spoken languages.

2.1. Sentiment analysis

Elouardighi et al. [11] presented a machine learning approach for Arabic SA for Facebook's comments, where *n*-grams, TF and TF-IDF were used as weighting schemes for features extraction, in which the results of SVM achieved better than NB and RF. Furthermore, Biltawi et al. [12] propose a hybrid approach for Arabic sentiment classification. The proposed approach combined the lexicon-based and the corpus-based approaches using Twitter and movies reviews dataset (OCR) [13]. The results revealed that the hybrid approach with RF and sixfold cross-validation performed better than the use of the corpus-based approach. Moreover, Daoud et al. [14] presented an approach for document clustering for three different news datasets. The approach depended on using *k*-means with particle swarm optimization (PSO) for the initial selection of clusters' seeds. The results showed that *k*-means based PSO with light stemmer achieved better results in regard to accuracy, *f*-measure, recall and precision. More interestingly, Al-Ayyoub et al. [15] provide a comprehensive survey for using deep learning for Arabic NLP tasks, such as optical character recognition, automatic speech recognition, caption generation, automatic machine translation, text categorization, dialect detection and segmentation, question answering, spam opinion detection and SA. Furthermore, Al-Azani and El-Alfy [16] combined the usage of textual features with Emojis for the sentiment classification of Arabic tweets (positive or negative). They implemented different features extraction techniques such as TF-IDF, BoW, latent semantic analysis and two forms of Word embedding. The results were superior for the combination of Emojis and skip-gram (as word embedding technique) when tested using SVM classifier based on *f*-measure. Overall, most of the studies have been utilising traditional machine learning algorithms with BoW, TF and TF-IDF weighting schemes as vectorization features for SA. However, Tubishat et al. [17] implemented an improved Whale optimization algorithm for feature selection of Arabic SA. The proposed algorithm improved the classification accuracy of Arabic sentiment classification.

The SA of the English language is widely studied throughout literature. Rout et al. [1] present a supervised and unsupervised sentiment and emotion analysis approach for unstructured social media data, in which the authors constructed a lexicon-based approach for the unsupervised approach and used the multinomial NB and SVM as a supervised approach, where the sentence-level classification with unigram and part-of-speech (PoS) features were the most accurate. Chaturvedi et al. [18] provided a SA review for categorising texts into factual or opinion, in which the authors categorised the subjectivity detection methods into syntactic and semantic models, whereas López et al. [19] argued that one of the major weaknesses of SA is the inability of the learning algorithm to generalise any input text type, which leads to the appearance of domain adaptation problem. Therefore, the authors claimed that the best solution is the proposal of ensemble SA methods. Thus, they proposed an evolutionary ensemble method that is tested on 13 different datasets. Tsakalidis et al. [20] present a thorough review of related Greek-SA resources, where the authors demonstrated the efficiency of resources for emotion and sarcasm detection. Vizcarra et al. [21] presented an approach for SA in the case of Spanish tweets, in which the model integrated several word embedding approaches alongside the convolutional neural networks (CNN), where it achieved very good results.

2.2. Hate speech detection

Hate speech has been encountered as a global problem; however, it is difficult to conclude a unified definition of it, owing to the various cultures, customs and traditions. In recent years, there have been some efforts and research studies that were initiated for addressing hate speech on online social media. Particularly in the Arabic context, there is a lack of conducted research studies that target hate speech on online social networks. However, Al-Hassan and Al-Dossari [22] presented a study on text mining approaches for addressing hate speech generally and presented challenges for addressing hate speech in Arabic context. In contrast, more studies have been performed for hate speech detection in the context of the English language. For instance, Chetty and Alathur [23] presented a detailed analysis of hate speech definition and other related terminologies as hate crime, terrorism, and extremism. Watanabe et al. [24] proposed a new method for detecting hate speech on Twitter, in which the authors used unigrams, sentimental, and semantic features to classify the tweets into binary and ternary classes. The results showed that the binary classification achieved better performance. Robinson et al. [25] examine the effect of feature selection for hate speech detection on Twitter. They found that the most informative features are the word and PoS *n*-grams, whereas Pitsilis et al. [26] proposed an ensemble of recurrent neural network classifiers for hate speech detection (racism and sexism) on Twitter. The results were outperforming the state-of-the-art methods. Zhang et al. [27] introduced a new algorithm based on a deep neural network that combined convolutional and gated recurrent networks for hate speech detection on Twitter, which focused on Islam and refugees as a case study where the algorithm achieved very good results. Similarly, an approach depending on CNN for hate speech detection on Twitter was designed by Biere and Bhulai [28].

Furthermore, a multi-task learning framework for the detection of hate speech and abusive language was adopted by Waseem et al. [29]. The model showed high generalisation ability for new datasets with regard to different contexts and cultures. Kshirsagar et al. [30] applied neural network algorithm for detecting online hate speech on Twitter. In addition, Unsvåg and Gambäck [31] studied the effects of user-related features on the performance of hate speech detection. They found that the most powerful features are the word-based features that accompanied with user-network features. Furthermore, a machine learning approach for detecting hate speech in the Indonesian language was presented by Alfina et al. [32], in which the authors extracted the word and character *n*-grams as features. The constructed dataset was used for training NB, SVM, logistic regression and RF and the results were very satisfying. Besides, Zhang and Luo [33] utilised a deep neural network model for extracting the semantic features related to hate speech. They evaluated their approach based on Twitter-collected dataset which performed better than other used algorithms based on *f*-score.

Nonetheless, an approach for Italian hate speech detection of the Facebook comments was proposed by Del Vigna et al. [34]. The collected dataset annotated by five annotators, with several extracted features: the morpho-syntactical features, the sentiment polarity, and the word embedding lexicons. The authors implemented SVM and Recurrent Neural Network (RNN)(long short-term memory) and the results were very effective.

In summary, the state of hate speech detection as a research direction stills rudimentary, yet in the Arabic context it is not investigated. Hence, from data mining and machine learning point of view, hate speech detection forms a rich resource for conducting more research studies and experiments.

3. Preliminaries

3.1. Natural Language Processing

NLP techniques handle abundant amount of data in order to extract useful knowledge by the utilisation of machine learning and data mining methods. However, in case of text, fundamental preprocessing steps should be applied for extracting numerical and statistical features from the text so it can be applied for machine learning algorithms. However, there are main concepts involved in any text preprocessing and structuring approach; this article will discuss the tokenization, stop words removal and text vectorization.

Tokenization is the process of dividing any piece of text into group of words or phrases which are known as tokens, in which the text is divided based on the presence of spaces. For instance, if sentence (S) equals to 'text mining is a new area of computer science', then tokenizing (S) results in S = ['text', 'mining', 'is', 'a', 'new', 'area', 'of', 'computer', 'science']. However, the existence of articles, conjunctions, prepositions and pronouns is insignificant since they do not express any specific meaning; these terms are known as the stop words. Therefore, removing the stop words from sentence (S) will results in S = ['text', 'mining', 'new', 'area', 'computer', 'science'].

Furthermore, transforming the unstructured text into structured text involves converting the piece of text into a vector of statistical features. Thus, if the tokens in sentence (S) is substituted by the number of times that each word appears in the text, then (S) will equal to [1,1,1,1,1,1]. This is known as text vectorization for feature extraction, which can be applied for machine learning algorithms. Basically, several NLP text preprocessing techniques are used mainly for feature extraction. The following subsections present the most common techniques, namely, BoW, TF and TF-IDF.

3.1.1. Bag-of-Words (BoW). BoW is a textual representational method used in several applications such as the classification models. In this representation, any piece of text (document, article or sentence) is considered as a collection of words with neither syntactical nor semantics relationships. It represents the occurrences of words within the text regardless of the words occurrence. For instance, having a collection of documents (D) that contains two documents $D = \{d_1, d_2\}$. Each document has a number of words or tokens n; so, $d_1 = \{w_{11}, w_{12}, ..., w_{1n_1}\}$ and $d_2 = \{w_{21}, w_{22}, ..., w_{2n_2}\}$, where n_1 is the number of words for d_1 and n_2 is the number of words for d_2, in which d is known as a document vector. For fixed-length documents vectors, all unique words (m) from all documents are extracted in order to create a vector of unique words of length m. Thus, each document vector is created based on the vector of unique words, where the presence of the words in the vector of unique words is represented by a binary value of (1), otherwise a value of (0).

Another close and well-known textual representational model is the TF, which is similar to the BoW approach. It differs in representing the frequency of the term in the corresponding piece of text and not the presence. The frequency is calculated by finding the number of occurrences of each keyword (n_k) in a document divided by the total number of keywords in the document as in equation (1) [35]

$$TF = \frac{n_k}{n} \tag{1}$$

3.1.2. Term Frequency-Inverse Document Frequency. Essentially, one of the major problems of TF is that the highest frequent words within a document are having the largest weights, where they might be insignificant, or less informative words, such as the word 'the'. Therefore, one of the evolved approaches to solve this problem is the TF-IDF approach that gives the rare words across all documents more weight than the frequent ones.

Where TF in TF-IDF is the frequency of the word in the current document, whereas the IDF is a measure of how rare the keyword is across all documents. In other words, the frequent words will have the numerical ratio of (N/N_k) with a value more close to 1.0, and consequently their TF-IDF value will approach 0.0 [35]. The formula of IDF is defined as in equation (2)

$$IDF = log_2\left(\frac{N}{N_k}\right) \tag{2}$$

In which, N is the total number of documents, and N_k is the number of documents that contain the keyword k. Whereas the TF-IDF is simply the product of TF and IDF, as in equation (3)

$$TF - IDF = \frac{n_k}{n}\ X\ log_2\left(\frac{N}{N_k}\right) \tag{3}$$

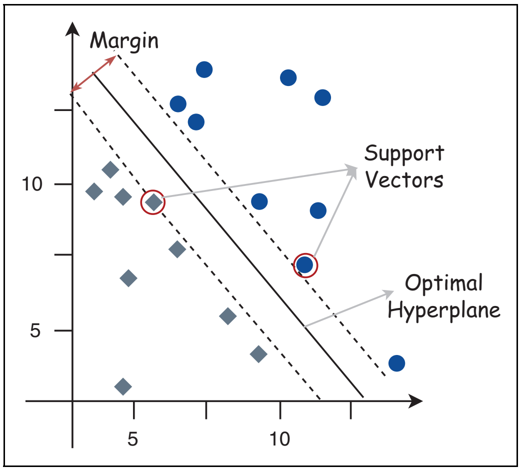

Figure 1. Illustration of SVM algorithm that utilises a hyperplane and the support vectors to best distinguish the classes.

3.2. Machine learning algorithms

This subsection discusses the machine learning algorithms that are used in this article which are SVM, NB, DT and RF.

3.2.1. SVM. SVM is a machine learning algorithm for classifying linear and non-linear data, where it uses non-linear methods to map the data into a higher dimension. It searches for the best boundary for classifying the data depending on the support vectors and the margins. The main idea is how to ideally fit the boundary to the training data using the support vectors for adjusting the boundary. The support vectors are data points in the search space that support the boundary.

Conventionally, the boundary is called a hyperplane that can be a linear line or a curve in two-dimensional search space. Finding the best hyperplane is done by looking for the boundary that best maximise the distance between the two classes beyond the boundary, where this distance is called the margin. However, most of the datasets are not linearly separable; hence, the hyperplane is assessed based on a penalty cost that is assigned to every data point existed on the margin region. Therefore, the non-linear data points are transformed into linear space by the utilisation of polynomial or Sigmoid functions [36,37].

Figure 1 shows how SVM finds the optimal hyperplane that separates the candidate classes by the aid of support vectors.

3.2.2. NB. NB algorithm is established based on Bayes theorem, which is a fundamental principle in statistics and probability theory. The classification does not rely on rules, but it rather depends on the probability and the conditional probability. Given a dataset of n attributes $\{a_1, a_2, ..., a_n\}$ and m data instances $\{d_1, d_2, ..., d_m\}$, with a set of classifications $\{c_1, c_2, ..., c_k\}$ and a set of probabilities $\{p_{c1}, p_{c2}, ..., p_{ck}\}$, the probability of a certain instance of a classification c_i is defined by equation (4) assuming that the attributes have values from $\{v_1, .., v_n\}$

$$p(c_i)\ X\ p(a_1 = v_1 \&\ a_2 = v_2\ \ldots\ \&\ a_n = v_n \mid c_i) \tag{4}$$

If the attributes are independent, then the probability will be calculated as in equation (5). Hence, this product is calculated for each instance based on all classes values, where the classification is chosen depending on the highest probability [38]

$$p(c_i)\ X\ p(a_1 = v_1 \mid c_i)\ X\ p(a_2 = v_2 \mid c_i)\ X\ \ldots\ X\ p(a_n = v_n \mid c_i) \tag{5}$$

3.2.3. DT. DT algorithm is used mainly for classification, where it constructs a flowchart or tree-like representation of the classification rules. Each internal node refers to an attribute, whereas the leaf nodes indicate the potential classification. To divide the data into different classes, the DT algorithm chooses the optimal attribute based on some measures such as the Gini index or the information gain. The algorithm stops depending on a predefined stopping criterion, such as reaching a maximum size of the tree [39].

3.2.4. RF. RF algorithm is an ensemble approach created by Breiman [40], which is a collection of DT classifiers that vote for the common class. Each DT encompasses different random set of attributes that were drawn from random vectors with a defined distribution. These sets of data are known as bootstrap samples. At classification, each tree votes for a class and the majority-voted class is chosen. The main difference of RF compared with the conventional DT is that RF does not

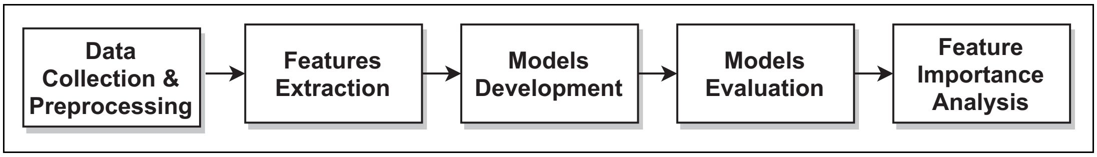

Figure 2. An overview flowchart of the methodology steps.

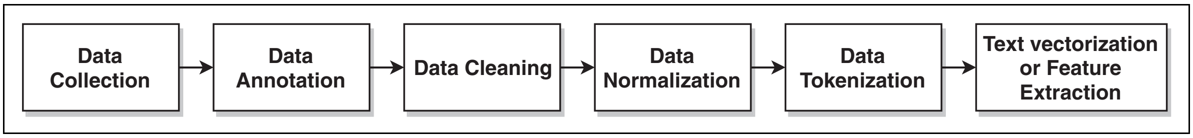

Figure 3. Steps of data collection and preprocessing.

Table 1. Description of collected tweets and their classes distributions.

Keyword	Translated Keyword	Tweets#	Positive (%)	Negative (%)
الوحدات	Alwahadat sport club	14	28.6	71.4
الدوري الاردني	The Jordanian league	44	13.6	86.4
الفيصلي الاردن	Faisaly Jordan	24	41.7	58.3
الاسلام والارهاب، تدمير الاسلام	Islam and terrorism, damage Islam	100	82.0	18.0
العنصرية	Racism	1193	46.8	53.2
لاجئ، لاجئوِن	Refugees	240	70.0	30.0
الحرية، الاعلام، الوطن، ناهض حتر، يساري متطرف	Freedom, media, homeland, Nahed Hattar, extreme	19	78.9	21.1

consider the whole set of variables at each split, but rather it looks for the best feature among the selected random subset of features, which leads to avoiding overfitting as well.

Typically, what is more intriguing of RF is its ability for measuring the variable importance by finding the amount of decrease of the error of prediction or classification across the development of the RF model [40].

4. Methodology

This section presents the proposed approach for tackling cyber hate speech detection on Twitter in the context of Arabic. As illustrated in Figure 2, it mainly starts by collecting and preprocessing the data (tweets), then is features extraction, Models development and training, afterwards is models' evaluation and feature importance analysis.

4.1. Dataset collection and preprocessing

Mainly, collecting and preprocessing the data, as shown in Figure 3, include the annotation of data, cleaning the data, normalisation, tokenization and feature extraction.

Basically, the collection of data is performed based on Twitter streaming Application Programming Interface (API) with 'rtweet' library,[1] and R programming language using RStudio framework [41]. The collection of data has targeted different areas like sport, religion, racism and journalism. The used keywords are as discussed in Table 1. The collected data accounts for 3696 tweets after removing duplicates and irrelevant retrieved tweets. The tweets are annotated and categorised into hate and non-hate depending mainly on two annotators. The annotators labelled the data independently from each other, and based on the overall meaning of the tweet if it has hate orientation or not. According to the overall collected dataset, the number of hate tweets corresponds to the positive class and equals to 843, whereas the number of non-hate tweets corresponds to the negative class and equals to 790, while the tweets that are neither hate nor non-hate (neutral) tweets account for 2061 tweets. In the following conducted experiments, the neutral tweets were excluded; therefore, the dataset is a combination of the positive and negative classes. The ratios of the two classes are described in Table 1.

A fundamental step in preprocessing the unstructured tweets is cleaning the data by filtering out non-Arabic characters, numbers, symbols, punctuation, hashtags, web addresses, diacritics and the Arabic stop words and negation words. Table 2 shows an example of the excluded negation and stop words, since the Natural Language Toolkit (NLTK) library [42] does not include all Arabic stop and negation words.

Table 2. An example of excluded Arabic, stop and negation words, and their translation into English.

Stop and negation keywords	Translated keywords
لما، عليك، أينما، ماذا، ممن، تلك، كأي	why, you, wherever, what, who, those, like,
، أنتم، بم، كيف، عدى، بيد، أن، حيث، ثم،	you, their, how, except, but, that, where, then,
سوف، هل، حتى، أقل، أكثر، نحو، كم، متى	will, do, up, less, more, towards, how many, when

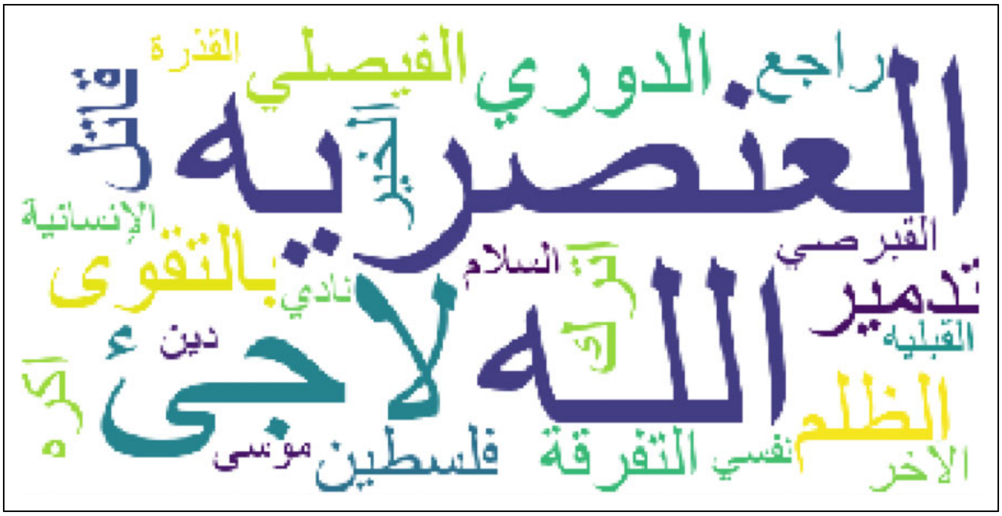

Figure 4. Representation of the most frequent words based on TF-IDF.

Following is to normalise the texts, which means converting different forms of Arabic characters into popular colloquial usage, for instance, the Alef Arabic character might be in several forms, indeed it should be converted from أ, إ, إِ, أُ into ا. Another example is converting the ي into ى. The tokenization of tweets is done based on NLTK library, which removes the punctuation from the tweet and divides the tweet into set of words based on the white-space delimiter.

Primarily, the retrieved data from Twitter includes a set of features that are related to user-profile features, which are the user identification number (ID), reply to certain user ID, does the tweet is retweeted, the tweet favourite count, the retweet count, the double retweet count, the retweet-friends count, the retweet-statuses count, the count of followers, the friends count, the listed-count, the count of statuses, the count of favourites and whether the account is verified. However, different features have been extracted from the collected datasets which are explained in the following subsection.

4.2. Features extraction

Various features have been extracted from the collected dataset, which are originally the user-profile features in addition to the words-based features, and the emotions-based features. The extracted word features are three sets of features: the BoW, TF, and TF-IDF models, which are implemented using Python programming language [43]. Figure 4 represents the most frequent words among all tweets based on TF-IDF model, in which the racism word is the most frequent word.

Similarly using Python, the emotion features extracted based on their Unicode which encompass wide range of emotion types, such as the happiness, sadness, or anger emotions, while the total extracted features accounts for 116. Several data combinations have been created depending on different features combinations. The combined collections of data are as follows.

- The word features solely, including either the BoW, the TF or the TF-IDF models.
- One set of the word features (BoW, TF, TF-IDF) and the profile features.
- One set of the word features (BoW, TF, TF-IDF) and the emotions features.
- The profile and emotions features.
- One set of the word features (BoW, TF, TF-IDF), besides the profile and emotion features.
- The profile features.
- The emotion features.

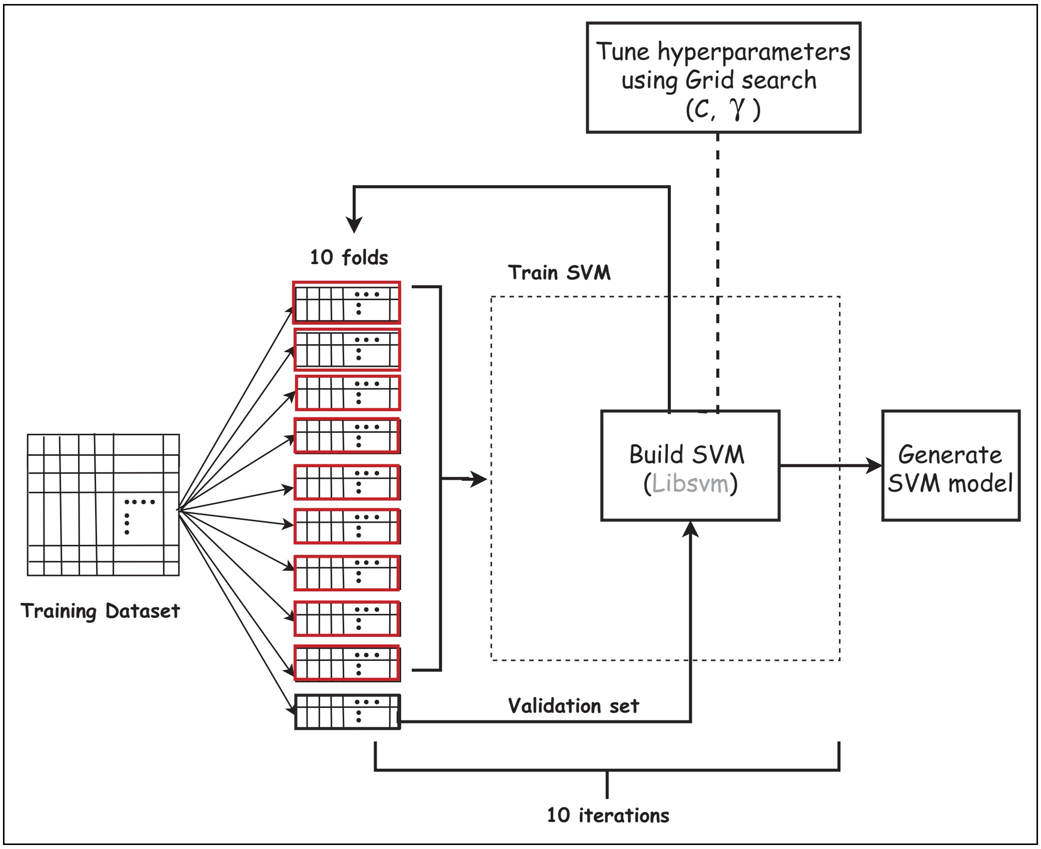

Figure 5. Development of SVM model.

4.3. Models development

Basically, all the algorithms that are used in this article which are the SVM, NB, DT, and RF were implemented using Python and the Scikit-learn machine learning package [44]. Certainly, the two fundamental processes to create a classification model from base machine learning algorithms are first preparing the training set of the data appropriately, and second is setting the parameters of the designated machine learning algorithms, which is an essential step to best boost the performance of the algorithms, and to avoid the problem of overfitting. The training sets used in this article involve different combinations of three sets of features (words, profile and emotions features) as described in the previous section, whereas setting and tuning the parameters of the algorithms are described in the following paragraphs.

Typically, SVM separates the data points into the desired classes by utilising a kernel function that specifies the type of the hyperplane. Hence, the kernel of the implemented SVM is a non-linear type of kernels, which is the radial basis function (RBF). Non-linear kernels require two main parameters: the gamma (γ) and the cost (C). The γ supports the algorithm to ideally fit the training data, while the cost parameter is a penalty parameter for enhancing the process of correctly categorising the training data points. Searching for the ideal combination of parameters' values results in having the optimal abilities of an algorithm.

Any search technique requires having an estimator (which is a classifier in case of classification), a search space, a search procedure, and a fitness function which represents the accuracy in classification. Herein, the Grid search procedure is utilised for searching the best values of γ and C. The Grid search uses a grid of a predetermined values of the selected parameters, to search comprehensively all the possible combinations of the parameters and to return the combination with the best accuracy of the SVM classifier [44]. The search space of γ is set to (0.01–0.50), and the cost is set to (0.01–1.00), where the Grid search resulted in the best value of the cost which equals 0.50 and the best value for gamma which equals 0.2.

Figure 5 shows the process of building the SVM model, which first starts searching the best hyperparameters of SVM, then divides the data into 10 folds for the cross-validation scheme for building the model.

NB algorithm classifies the data depending on the probability theory of Bayes, where it treats the features independently from each other. However, since it relies on the probability of classifying the data, it does not need any parameters for tuning. The utilised NB is the Gaussian NB which is mainly best suited with the continuous variables. Herein, the

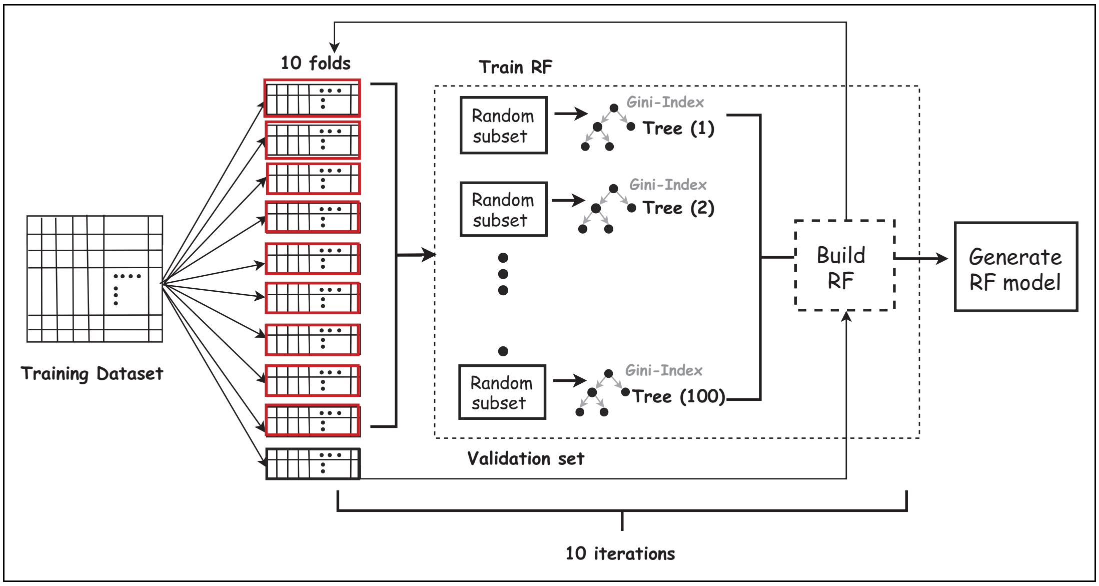

Figure 6. Development of RF model.

Gaussian NB implemented based on equation (6), in which the standard deviation (σ_y) and the mean (μ_y) are based on the maximum probability [44].

Similarly, the NB model is created based on 10-fold cross-validation scheme

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{6}$$

Regarding the DT algorithm, DT is non-parametric algorithm that uses some learned decision rules to classify the data, in which the decision rules are in tree-like structure. For building the tree, the splitting criterion for partitioning the data is based on Gini index, which performs such ranking for the features based on the Gini index score, where the feature with the best score is selected as a splitting point. Besides, another important parameter is the maximum depth of the tree, which is set to the state that all nodes are extended until they are all pure.

The RF classifier is a collection of DTs that use different subsets of the original training data, which is drawn with replacement. The main advantage of the RF is avoiding the potential of overfitting. RF has several parameters to be set such as the number of trees, which is set to 100 trees, with Gini index splitting criterion. The number of features in each random subset is set as the square root value of the total number of features. As well as, the maximum depth of the tree is set to be extended until all leaves are pure. Figure 6 shows the development of RF classifier model based on DT and Gini index splitting criterion all throughout 10-fold cross-validation [44].

4.3.1. Experimental setup. Basically, the experiments are conducted using Spyder development framework, version 3.7 [45]. The experiments are performed on Desktop Computer with Windows 7 operating system, memory of 16 Gigabyte, and the processor is Intel Core i7 with 3.4 Gigahertz. The dataset is divided into training and testing based on 10-fold cross-validation. Figure 7 represents a graphical representation of the methodology, which starts by collecting and processing the data alongside extracting the sets of features of words, profile, and emotions. Afterwards is creating different 15 combinations of the features, which is followed by applying the machine learning classifiers of SVM, NB, DT and RF. Then is evaluating the performance of the classification models, and investigating the features with the highest predictive ability of the target class (hate or non-hate).

4.4. Evaluation and assessment

The used evaluation measures are accuracy, precision, recall, and *g*-mean, which are calculated using the confusion matrix, where the True Positives (TP) are the tweets that are predicted as hate and they are actually hate, the True Negatives (TN) are the tweets that are predicted as non-hate and they are actually non-hate, the False Positives (FP) represents the tweets that are predicted as hate, but they are actually non-hate, and the False Negatives (FN) are the tweets that are predicted as non-hate, but they are actually hate. The used metrics are defined as follows.

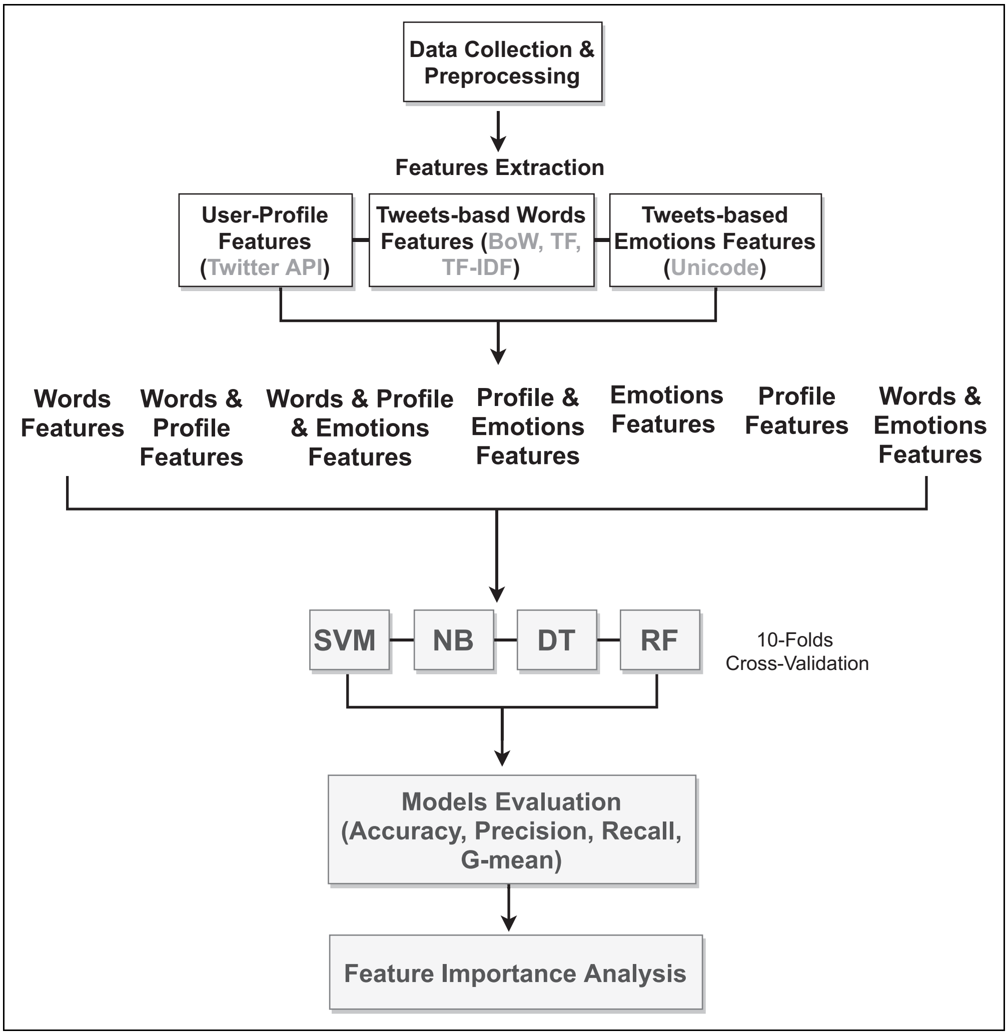
Data Collection & Preprocessing
Features Extraction
User-Profile Features (Twitter API)
Tweets-basd Words Features (BoW, TF, TF-IDF)
Tweets-based Emotions Features (Unicode)
Words Features
Words & Profile Features
Words & Profile & Emotions Features
Profile & Emotions Features
Emotions Features
Profile Features
Words & Emotions Features
SVM
NB
DT
RF
10-Folds Cross-Validation
Models Evaluation (Accuracy, Precision, Recall, G-mean)
Feature Importance Analysis

Figure 7. An overview of the methodology design.

Accuracy: which is the ratio of correctly classified hate and non-hate instances over all the correct and the incorrect number of classified instances. Equation (7) is the equation for accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision: also known as positive predictive value, which is identified as the ratio of the tweets that correctly identified as hate over the number of all hate (positive) tweets, which is represented as in equation (8)

$$Precision = \frac{TP}{FP + TP} \tag{8}$$

Recall: known as sensitivity, which means how much the classifier can recognise the positive instances (hate) of tweets, which is defined by equation (9)

$$Recall = \frac{TP}{FN + TP} \tag{9}$$

G-mean: known as the geometric mean, which indicates the balance of classification performances on the hate (positive class) and the non-hate (the negative class), as given by equation (10)

$$G - mean = \sqrt{Recall(class1).Recall(class2)} \tag{10}$$

4.5. Feature importance analysis

One of the most eminent characteristics of RF is its ability of performing a feature ranking, since it utilises feature importance measures such as the Gini index or the information gain. This section analyzes the most influential features for spotting hate speech on Twitter.

Gini index or Gini impurity is a measure used to promote the splitting at each node of the tree. Since Gini index indicates the impurity of a dataset, it is used to find the features that decrease the mean of weighted impurity over all trees. Hence, the features with least Gini index or best maximised mean decrease in impurity (MDI) are the most informative features for predicting the target class. For any dataset (*d*), the Gini index is defined as in equation (11), where *m* is the total number of classes, p_i is the probability that a data instance in *d* holds class C_i [37]

$$Gini(d) = 1 - \sum_{i=1}^{m} p_i^2 \tag{11}$$

The decrease in impurity ($\Delta Gini(J)$) for any feature (*J*) is defined by equation (12) as the difference between the Gini index of the dataset *d* and the Gini index of the dataset *d* at certain feature *j*. The $Gini_J(d)$ is based on a binary split that results in dividing dataset *d* into two parts: d_1 and d_2, and calculated as in equation (13) [37], while the mean decrease in impurity is calculated based on all produced trees of the RF

$$\Delta Gini(J) = Gini(d) - Gini_J(d) \tag{12}$$

$$Gini_J(d) = \frac{|d_1|}{|d|} Gini(d_1) + \frac{|d_2|}{|d|} Gini(d_2) \tag{13}$$

5. Experiments and results

Mainly, the collected dataset is categorised into 15 combinations of data that are applied to several machine learning models for predicting hate. Second is analysing the most crucial features for identifying hate at tweets based on RF classifier.

5.1. Performance results of classifiers

Table 3 represents the classification performance of the classifiers (SVM, NB, DT, and RF) as shown per row, alongside the performance metrics (accuracy, recall, precision and *g*-mean) as per column, all at different combinations of words, profile and emotions features.

Considering the BoW model of word features, it is noticeable that RF performs better than other classifiers in terms of accuracy and *g*-mean when the word features accompanied with either the profile features, or with emotion features, or even with both the profile and emotion features. Whereas using merely the BoW words features, NB performed better than others with slightly close performance to RF at the other combinations. Generally speaking, the best performance is achieved when all features combined together by the RF reaching a peak of 0.910 in terms of accuracy and *g*-mean, while achieving 0.923 in terms of recall and 0.902 in terms of precision.

On the other hand, looking at the TF set of features, NB outperformed other algorithms at most of the combinations; at TF word features alone by holding best accuracy of 0.894; at TF with profile features by having the accuracy equals to 0.887; and at TF with emotion features, where the accuracy is 0.904. However, the RF achieved best overall when combined the three sets of features (words (TF), profile, and emotion) by having accuracy equals to 0.898, and *g*-mean equals to 0.897. Even that, the RF accomplished slightly better than other algorithms in case of BoW instead of TF, with the three sets of features.

Whereas in case of TF-IDF, NB achieved better in the case of words alone (TF-IDF) by having accuracy equals to 0.894, and in the case of the words with emotion features by having accuracy equals to 0.888. While the RF achieved best in two cases: first is the case of word and profile features, which obtained superior accuracy of 0.913 and g-mean of 0.912. Second is the case of TF-IDF & profile & emotions, in which the accuracy equals to 0.890 and *g*-mean equals to 0.886.

Nonetheless, in case of the profile set of features alone, the RF performed the best in terms of accuracy, which is 0.885, the precision is 0.907, and the *g*-mean is 0.883. Where it is approximately close to NB performance at TF-IDF with the emotion features, and to the case of NB at TF with profile features. At the case of emotion features alone, the performance of the algorithms declined considerably and approximately to the half when compared with the performance of algorithms at the previous combinations. RF obtained the maximum performance having the accuracy of 0.592.

Table 3. Comparison of classification performances over all data combinations for all classifiers.

Word features	Profile features	Emotion features	Classifier	Accuracy	Recall	Precision	*G*-mean
BoW	–	–	SVM	0.842	**0.978**	0.786	0.823
			NB	**0.893**	0.848	**0.935**	**0.892**
			DT	0.877	0.915	0.858	0.874
			RF	0.884	0.929	0.860	0.880
BoW	✓	–	SVM	0.853	**0.979**	0.796	0.840
			NB	0.891	0.883	**0.894**	0.890
			DT	0.872	0.870	0.873	0.872
			RF	**0.897**	0.942	0.867	**0.896**
BoW	–	✓	SVM	0.858	**0.951**	.812	0.849
			NB	0.887	0.835	**0.932**	0.885
			DT	0.879	0.929	0.851	0.875
			RF	**0.895**	0.889	0.903	**0.895**
BoW	✓	✓	SVM	0.836	0.886	0.814	0.830
			NB	0.908	**0.929**	0.892	0.908
			DT	0.895	0.911	0.884	0.895
			RF	**0.910**	0.923	**0.902**	**0.910**
TF	–	–	SVM	0.839	**0.991**	0.777	0.817
			NB	**0.894**	0.893	**0.896**	**0.894**
			DT	0.858	0.905	0.836	0.856
			RF	0.871	0.937	0.841	0.865
TF	✓	–	SVM	0.852	**0.999**	0.791	0.824
			NB	**0.887**	0.883	**0.901**	**0.887**
			DT	0.876	0.913	0.865	0.871
			RF	0.885	0.959	0.851	0.875
TF	–	✓	SVM	0.850	**0.955**	0.807	0.840
			NB	**0.904**	0.904	**0.900**	**0.904**
			DT	0.879	0.898	0.866	0.878
			RF	0.890	0.879	0.897	0.889
TF	✓	✓	SVM	0.860	**0.932**	0.829	0.853
			NB	0.892	0.870	**0.908**	0.890
			DT	0.882	0.893	0.873	0.882
			RF	**0.898**	0.915	0.887	**0.897**
TF-IDF	–	–	SVM	0.839	**0.953**	0.788	0.830
			NB	**0.894**	0.867	**0.910**	**0.892**
			DT	0.874	0.879	0.866	0.874
			RF	0.888	0.860	0.904	0.885
TF-IDF	✓	–	SVM	0.875	**0.976**	0.825	0.866
			NB	0.895	0.857	**0.926**	0.893
			DT	0.882	0.885	0.882	0.882
			RF	**0.913**	0.941	0.897	**0.912**
TF-IDF	–	✓	SVM	0.843	**0.915**	0.812	0.837
			NB	**0.888**	0.872	0.891	**0.887**
			DT	0.868	0.875	0.856	0.868
			RF	0.884	0.844	**0.908**	0.880
TF-IDF	✓	✓	SVM	0.844	**0.980**	0.783	0.828
			NB	0.873	0.844	**0.897**	0.871
			DT	0.871	0.894	0.859	0.870
			RF	**0.890**	0.951	0.854	**0.886**
–	✓	–	SVM	0.584	**0.899**	0.543	0.503
			NB	0.577	0.898	0.539	0.492
			DT	0.864	0.855	0.865	0.863
			RF	**0.885**	0.850	**0.907**	**0.883**
–	–	✓	SVM	0.563	0.987	0.534	0.360
			NB	0.579	**0.995**	**0.544**	0.390
			DT	0.564	0.993	0.552	0.427
			RF	**0.592**	0.993	0.552	**0.428**
–	✓	✓	SVM	0.617	0.612	0.620	0.602
			NB	0.579	**0.967**	0.549	0.408
			DT	0.871	0.872	0.873	0.870
			RF	**0.882**	0.864	**0.900**	**0.882**

SVM: Support Vector Machine; NB: Naive Bayes; DT: Decision Tree; RF: Random Forest; TF: Term Frequency; TF-IDF: Term Frequency-Inverse Document Frequency. Bold indicates best results among all methods.

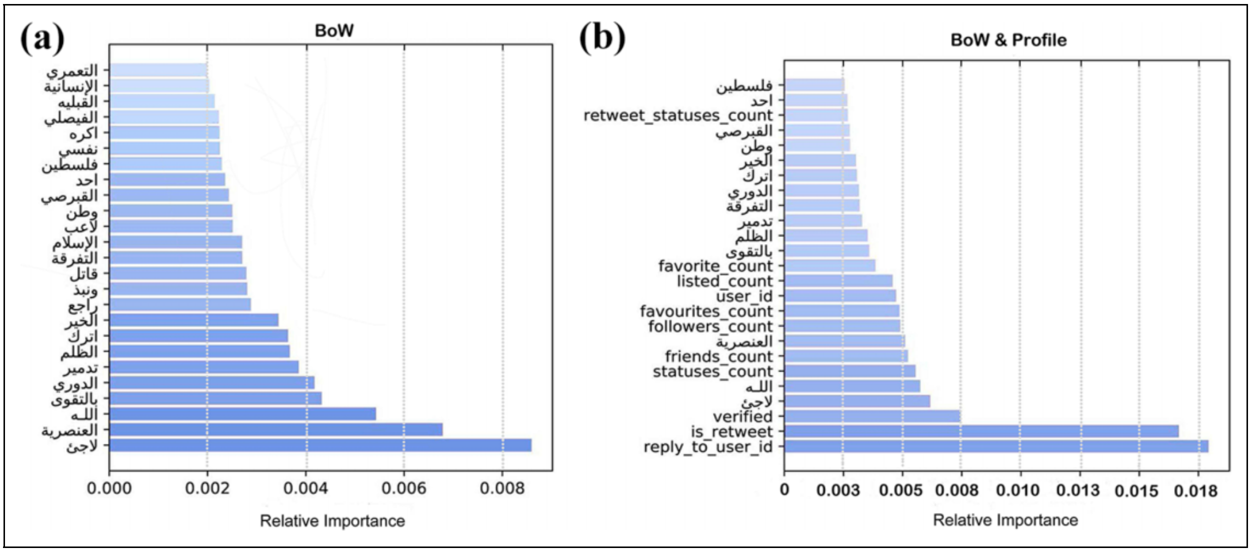

Figure 8. Analysis of features importance for (a) Bow and (b) Bow with profile features.

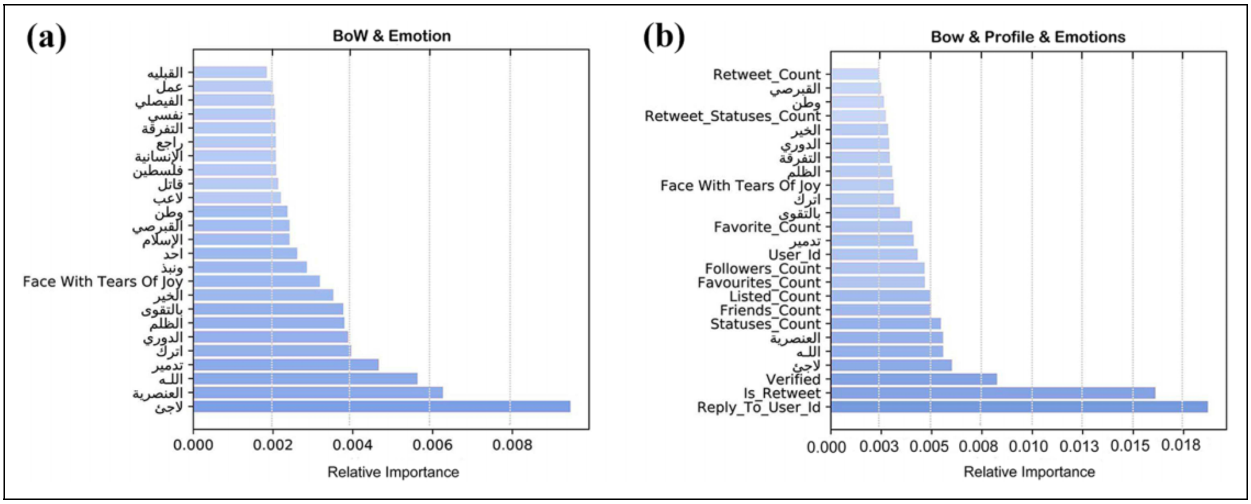

Figure 9. Analysis of features importance for (a) Bow with emotions features and (b) Bow with profile, and emotions features.

Yet in the last case, the case of profile and emotion features, RF outperformed other algorithms by achieving the best accuracy equals to 0.882, and similarly is the *g*-mean equals to 0.882, while the precision is 0.900.

Overall, the best result is achieved by RF at TF-IDF with profile features.

5.2. Analysis of the most relevant features

Figures 8–15 analyse the first 25 most important features that have the minimum MDI value depending on RF classifier. The horizontal-axis represents the relative importance of the features which is indicating the mean decrease of impurity, where the decrease of impurity is defined previously by equation (12), in Section 4. While the vertical-axis represents the names of the features. From the previous section, it has been concluded that RF was superior in mainly two cases: first is at TF-IDF with the profile features, and second is at BoW when it accompanied with all features. As RF can achieve quite high performance results, interpreting its assignment of the importance of the features might unleash the potential for better feature selection and better optimization the performance.

Regarding the words features alone (Bow, TF or TF-IDF) as indicated by Figures 8(a), 10(a), 12(a), the three features of racism, God and emigrant ('العنصريه، الله، لاجئ') were the first best three features in the same order, except in the case of BoW it was as emigrant, racism and God ('لاجئ، العنصريه، الله'). Nonetheless it is similar, too, to the combination of BoW

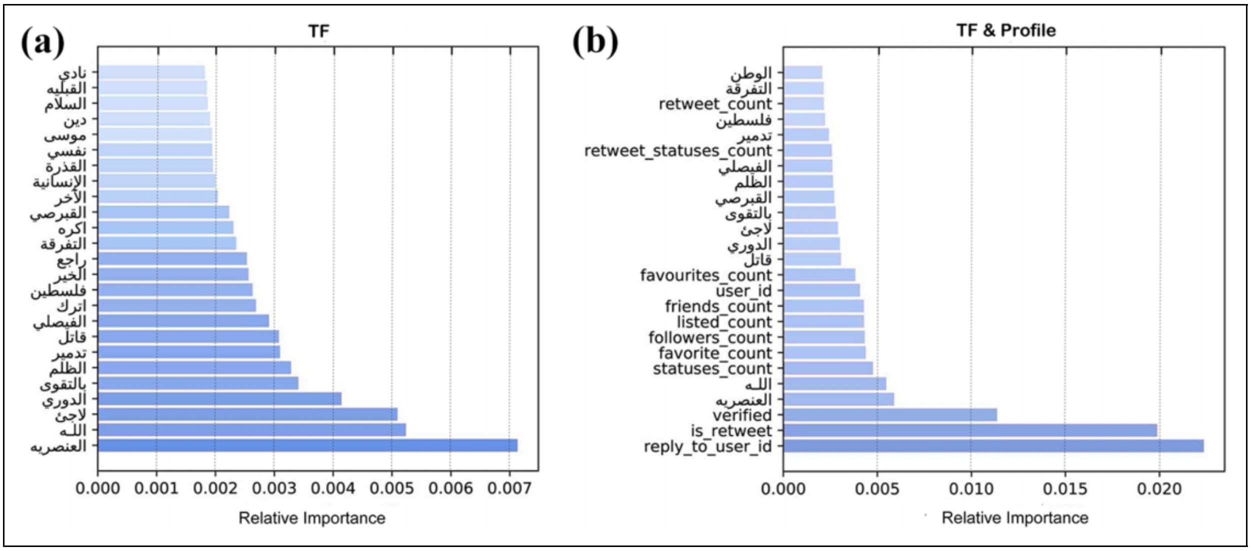

Figure 10. Analysis of features importance for (a) TF and (b) TF with profile features.

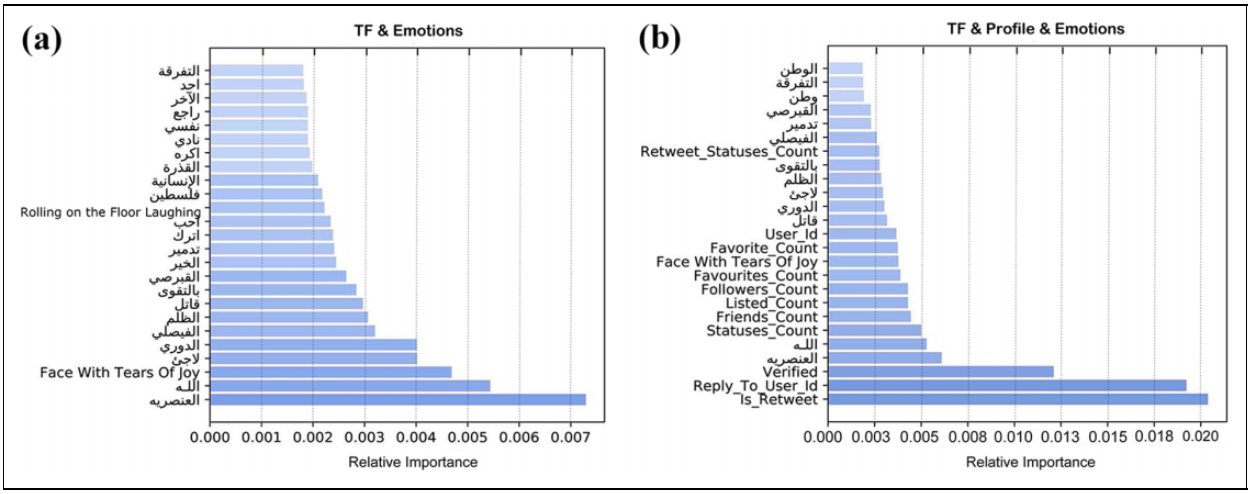

Figure 11. Analysis of features importance for (a) TF with emotions features and (b) TF with profile, and emotions features.

and emotions in Figure 9(a). On the contrary, the three sets of words features (Bow, TF, or TF-IDF) had the last (25th) important feature a word that is related to sport, where in case of Bow (Figure 8(a)) it was referring to a famous soccer player 'التعمري', and for the case of TF and TF-IDF, it was a sport club 'نادي'.

In addition, Figure 12(b) shows the most important features for TF-IDF with profile features, which exhibits that 'reply to user id' is the most important feature with the minimum MDI value, while next to it are the ('is retweeted', 'if verified', and racism 'العنصريه'). However, homeland (الوطن) was the least important feature (the 25th) which has the maximum MDI. Furthermore, a very close set of features were achieved by the combinations of features of (TF & profile) in Figure 10(b), (TF & profile & emotions) in Figure 11(b), and (TF-IDF & profile & emotions) in Figure 13(b).

Whereas, for the BoW with all features (profile and emotions) as in Figure 9(b), it obtained similarly the same first three features as (TF-IDF with profile), which are ('reply to user id', 'is retweeted' and 'if verified'), but close to them is the feature emigrant ('لاجئ'). Yet, 'retweet count' associated with the highest MDI value. Likewise is the BoW with profile features in Figure 8(b), except that the feature Palestine (فلسطين) is the last feature with the highest MDI value.

Relating to the emotions features as illustrated by Figure 14(b), it is obvious that 'face with tears of joys' and 'rolling on the floor laughing' are the most occurring types of emotions. While in regard to the profile features (Figure 14(a)), the features of ('friends count', 'favourites count' and 'reply to user id') were the most important features. While in contrast, combining the profile features with TF-IDF & emotions (Figure 13(a)) resulted in that the ('is retweeted', 'and reply to

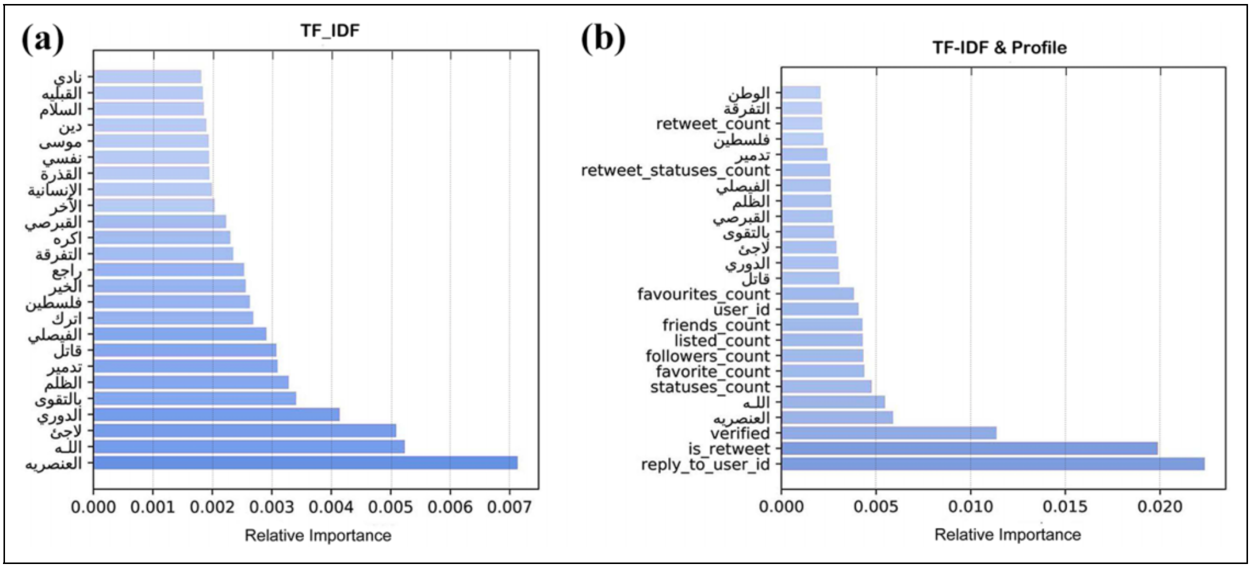

Figure 12. Analysis of features importance for (a) TF-IDF features and (b) TF-IDF with profile features.

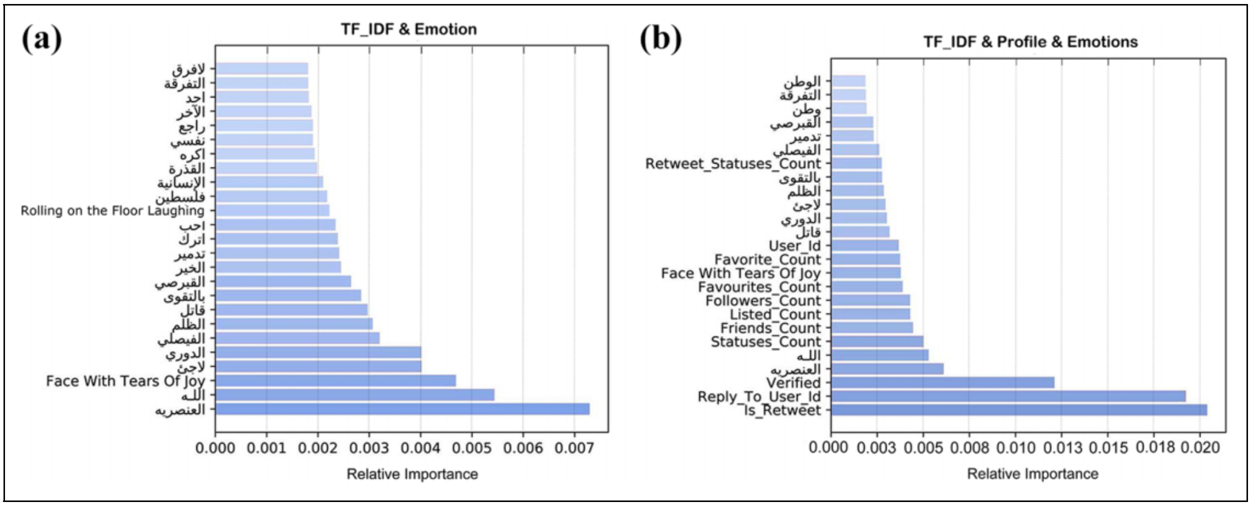

Figure 13. Analysis of features importance for (a) TF-IDF with emotions features and (b) TF-IDF with profile, and emotions features.

user id') were the most important features, similarly with TF-IDF without the emotions. In addition, it is the same for the profile with TF & emotion or the TF alone (Figures 11(b) and 10(a)), respectively, and the Bow alone (Figure 8(a)), or the Bow & emotion (Figure 9(a)), as well.

Figure 15 shows the most important features of the combination of profile and emotions. It is noticeable that ('friends count', 'favourites count', and 'reply to user id') are the superior features among all, whereas the discrepancy here is that the MDI value of 'is retweet' feature has raised up considerably to approximately the half. Generally based on the obtained results, the features of racism, emigrant, and God were the most keywords features that indicated the speeches of hate.

6. Conclusion and future works

Cyber hate speech is a serious problem world-wide that endangers the cohesion of civil societies. The continuous evolution of computational technologies has opened the door for novel smart technologies to emerge. Hence, machine learning is one of the contemporary and cutting-edge solutions for many problems. This article proposed a machine learning-based approach for addressing the problem of cyber hate speech of the Arabic context over Twitter. The data collected using Twitter streaming API were processed and deployed into machine learning algorithms (SVM, NB, DT, and

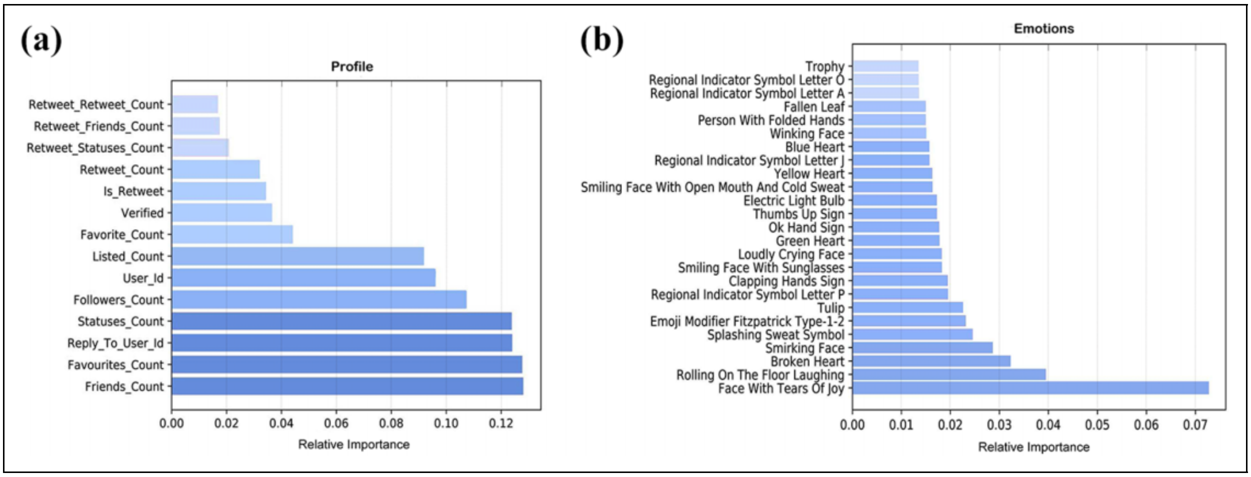

Figure 14. Analysis of features importance for (a) profile features and (b) emotions features.

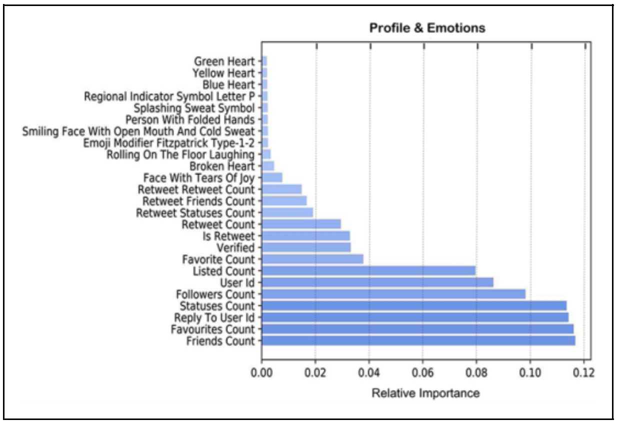

Figure 15. Analysis of features importance for emotions and profile features.

RF) using Python and Spyder development framework. Overall, RF classifier was performing the best over other classifiers, while interpreting the most influential features that accompanied with hate tweets; they were the racism, emigrant, and the God word-based features.

However, despite the promising obtained results there are still challenges, limitations and further research niches to interpret and explore. Hate as a term is subjective and can be expressed in a wide range of areas not restricted to the sport, religious or racial issues. In consequence, preventing it from the Internet and mainly from social networks becomes a challenge with the ever-increasing technologies and platforms, where people are more opened to freely express, discuss and debate without any responsibilities. Evidently, there is a lack of benchmark datasets especially in Arabic that are designed for hate speech detection and prevention in various areas. Also, handling Arabic text is not trivial, where there are numerous Arabic dialects yet the use of colloquial language. Inspecting the colloquial language alone is a major challenge since social media users write in different forms, while further the misspelling problem for a highly morphological language. Although there are some efforts for flourishing the Arabic context in machine learning, still the available Arabic resources are poor. Nonetheless, exploring the semantic meaning of expressions is very interesting and crucial for effective detection of hatred much more than merely depending on statistical vectorization techniques. Even that dealing with The Arabic textual data extremely relies on the preprocessing step; however, machine learning realm is wide and advances every day. As with the increasing emergence of deep learning, it is a massive interest to investigate its behaviour and efficiency in dealing with large textual data for the prediction of Arabic cyber hate speech.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by the Deanship of Scientific Research in the University of Jordan, Amman, Jordan.

ORCID iD

Hossam Faris https://orcid.org/0000-0003-4261-8127

Note

1. rtweet: https://rtweet.info/

References

[1] Rout JK, Choo K-KR, Dash AK, et al. A model for sentiment and emotion analysis of unstructured social media text. *Electron Commerce Res* 2018; 18(1): 181–199.

[2] Boudad N, Faizi R, Haj Thami RO, et al. Sentiment analysis in Arabic: a review of the literature. *Ain Shams Eng J* 2017; 9(4): 2479–2490.

[3] European Court. European court of human rights, https://www.echr.coe.int (accessed July 2019).

[4] Facebook Team. Facebook Community Standards, https://web.facebook.com/communitystandards/hate_speech (accessed 12 July 2019).

[5] Twitter Team. Twitter rules and policies – hateful conduct policy, https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy (accessed June 2019).

[6] Abu-Taieh E, Alfaries A, Al-Otaibi S, et al. Cyber security crime and punishment: comparative study of the laws of Jordan, Kuwait, Qatar, Oman, and Saudi Arabia. *Int J Cyb War Terr* 2018; 8(3): 46–59.

[7] Jordanian Ministry Jordanian ministry of justice, http://www.moj.gov.jo/EchoBusV3.0/SystemAssets/5d38ea27-5819-443e-a380-b65c7e1f5b56.pdf (accessed July 2019).

[8] Statista Inc. The most common languages on the internet, https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet (accessed July 2019).

[9] Badaro G, Baly R, Hajj H, et al. A survey of opinion mining in Arabic: a comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations. *ACM Trans Asian Low Res Lang Inf Process* 2019; 18(3): 27.

[10] Biltawi M, Etaiwi W, Tedmori S, et al. Sentiment classification techniques for Arabic language: a survey. In: *2016 7th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 5–7 April 2016, pp. 339–346. New York: IEEE.

[11] Elouardighi A, Maghfour M, Hammia H, et al. A machine learning approach for sentiment analysis in the standard or dialectal Arabic Facebook comments. In: *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, 24–26 October 2017, pp. 1–8. New York: IEEE.

[12] Biltawi M, Al-Naymat G and Tedmori S. Arabic sentiment classification: a hybrid approach. In: *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 11–13 October 2017, pp. 104–108. New York: IEEE.

[13] Mass AL, Daly RE, Pham PT, et al. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, 19–24 June 2011, pp. 142–150. Philadelphia, PA: Association for Computational Linguistics.

[14] Daoud AS, Sallam A and Wheed ME. Improving Arabic document clustering using K-means algorithm and particle swarm optimization. In: *2017 Intelligent Systems Conference (IntelliSys)*, London, 7–8 September 2017, pp. 879–885. New York: IEEE.

[15] Al-Ayyoub M, Nuseir A, Alsmearat K, et al. Deep learning for Arabic NLP: a survey. *J Comput Sci* 2018; 26: 522–531.

[16] Al-Azani S and El-Alfy E-SM. Combining emojis with Arabic textual features for sentiment classification. In: *2018 9th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 3–5 April 2018, pp. 139–144. New York: IEEE.

[17] Tubishat M, Abushariah MAM, Idris N, et al. Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Appl Intell* 2019; 49(5): 1688–1707.

[18] Chaturvedi I, Cambria E, Welsch RE, et al. Distinguishing between facts and opinions for sentiment analysis: survey and challenges. *Inform Fusion* 2018; 44: 65–77.

[19] López M, Valdivia A, Martnez-Cámara E, et al. E2SAM: evolutionary ensemble of sentiment analysis methods for domain adaptation. *Inf Sci* 2019; 480: 273–286.

[20] Tsakalidis A, Papadopoulos S, Voskaki R, et al. Building and evaluating resources for sentiment analysis in the Greek language. *Lang Res Eval* 2018; 52(4): 1021–1044.

[21] Vizcarra G, Mauricio A and Mauricio L. A deep learning approach for sentiment analysis in Spanish tweets. In: *International Conference on Artificial Neural Networks*, Rhodes, 4–7 October 2018, pp. 622–629. Cham: Springer.

[22] Al-Hassan A and Al-Dossari H. Detection of hate speech in social networks: a survey on multilingual corpus. *Comp Sci Inf Tech* 2019; 9(2): 83.

[23] Chetty N and Alathur S. Hate speech review in the context of online social networks. *Aggress Viol Behav* 2018; 40: 108–118

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

[25] Robinson D, Zhang Z and Tepper J. Hate speech detection on Twitter: feature engineering vs feature selection. In: *European Semantic Web Conference*, Heraklion, 3–7 June 2018, pp. 46–49. Basel: Springer.

[26] Pitsilis GK, Ramampiaro H and Langseth H. Effective hate-speech detection in Twitter data using recurrent neural networks. *Appl Intell* 2018; 48(12): 4730–4742.

[27] Zhang Z, Robinson D and Tepper J. Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In: *European Semantic Web Conference*, Heraklion, 3–7 June 2018, pp. 745–760. Basel: Springer.

[28] Biere S and Bhulai S. *Hate speech detection using natural language processing techniques*. PhD thesis, Vrije Universiteit Amsterdam, 2018.

[29] Waseem Z, Thorne J and Bingel J. Bridging the gaps: multi task learning for domain transfer of hate speech detection. In: J Golbeck (ed.) *Online harassment*. Berlin/Heidelberg, Germany: Springer, 2018, pp. 29–55.

[30] Kshirsagar R, Cukuvac T, McKeown K, et al. Predictive embeddings for hate speech detection on Twitter. arXiv preprint: arXiv: 1809.10644, 2018.

[31] Unsvåg EF and Gambäck B.The effects of user features on twitter hate speech detection. In: *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, Brussels, 31 October 2018, pp. 75–85. Association for Computational Linguistics.

[32] Alfina I, Mulia R, Fanany MI, et al Hate speech detection in the Indonesian language: a dataset and preliminary study. In: *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Bali, Indonesia, 28–29 October 2017, pp. 233–238. New York: IEEE.

[33] Zhang Z and Luo L. Hate speech detection: a solved problem? The challenging case of long tail on Twitter. arXiv preprint: arXiv: 1803.03662, 2018.

[34] Del Vigna F, Cimino A, Dell'Orletta F, et al. Hate me, hate me not: hate speech detection on Facebook. In: *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, Venice, 17–20 January 2017.

[35] Kotu V and Deshpande B. *Data science: concepts and practice*. Burlington, MA: Morgan Kaufmann, 2018.

[36] Suykens JAK and Vandewalle J. Least squares support vector machine classifiers. *Neur Process Lett* 1999; 9(3): 293–300.

[37] Han J, Pei J and Kamber M. *Data mining: concepts and techniques*. Waltham, MA: Elsevier, 2011.

[38] Domingos P and Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach Learn* 1997; 29(2–3): 103–130.

[39] Quinlan JR. Induction of decision trees. *Mach Learn* 1986; 1(1): 81–106.

[40] Breiman L. Random forests. *Mach Learn* 2001; 45(1): 5–32.

[41] Verzani J. *Getting started with RStudio*. Sebastopol, CA: O'Reilly Media, Inc., 2011.

[42] Loper E and Bird S. NLTK: the natural language toolkit. In: *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, Barcelona, Spain, 2004, pp. 214–217. Philadelphia, PA: Association for Computational Linguistics.

[43] Van Rossum G and Drake FL, Jr. *Python tutorial*. Amsterdam: Centrum Wiskunde & Informatica, 1995.

[44] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011; 12: 2825–2830.

[45] Raybaut P. Spyder: scientific Python development environment, 2009, https://github.com/spyder-ide/spyder (accessed 2017).