Computing (2020) 102:501–522
https://doi.org/10.1007/s00607-019-00745-0

Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach

Zafer Al-Makhadmeh[1] · **Amr Tolba**[1,2]

Received: 2 April 2019 / Accepted: 24 July 2019 / Published online: 1 August 2019

Abstract
Over the last decade, the increased use of social media has led to an increase in hateful activities in social networks. Hate speech is one of the most dangerous of these activities, so users have to protect themselves from these activities from YouTube, Facebook, Twitter etc. This paper introduces a method for using a hybrid of natural language processing and with machine learning technique to predict hate speech from social media websites. After hate speech is collected, steaming, token splitting, character removal and inflection elimination is performed before performing hate speech recognition process. After that collected data is examined using a killer natural language processing optimization ensemble deep learning approach (KNLPEDNN). This method detects hate speech on social media websites using an effective learning process that classifies the text into neutral, offensive and hate language. The performance of the system is then evaluated using overall accuracy, f-score, precision and recall metrics. The system attained minimum deviations mean square error −0.019, Cross Entropy Loss −0.015 and Logarithmic loss L-0.0238 and 98.71% accuracy.

Keywords Social media · YouTube · Facebook · Twitter · Hate speech · Killer natural language processing optimizing ensemble deep learning approach

Mathematics Subject Classification 01-00 · 01-02 · 11Axx · 03-04 · 03Bxx · 39-00

✉ Zafer Al-Makhadmeh
zalmakhadmee@ksu.edu.sa

[1] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

[2] Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-Kom 32511, Egypt

1 Introduction

The enormous growth of technology has increased activity on social media platforms [1], including using of Twitter, Facebook, Instagram to communicate. Most people use their Twitter accounts to follow people, to participate in social activities, and to convey their opinions through comments. During these digital exchanges, it is easy for people to use hateful or negative speech; this is because the anonymity of social media allows people to feel more comfortable and safe participating in discriminating activities [2]. Hate speech is defined statements that discriminate against individuals or groups of people based on characteristics such as gender, race, ethnicity, skin color, nationality, political activity, sexual orientation or region characteristics. These hateful activities [3] occur all over the world; for example, in the United Kingdom [4], hates speech activities are directed at Muslim and migrant people; the country's exit from the European Union (EU), dubbed "Brexit"; as well as the London and Manchester attacks. Around 80% of people in hate speech based on gender, religion, and ethnicity. In the United States, the Donald Trump election campaign, in 2015–2016 also incited numerous hate speech activities. In India, a sudden Pakistani attack on Indian army forces in 2019 also precipitated negative or hateful speech.

These hateful activities are spread via social media platforms [5] such as Facebook, Twitter, Instagram and YouTube, and addressing or preventing such speech presents a serious new problem for society. Online hate has a sufficient impact that it warrants significant countermeasures, and employing such measures successfully requires new methods for tracking online hate speech. YouTube, Facebook, Twitter, Instagram companies have spent millions of US dollars to track these negative activities, which is a testament to how seriously they take this issue [6]. However, the current available countermeasures require considerable effort to locate and delete negative speech. In addition, extant tracking processes are not suited to real time applications, are labor-intensive, and consume considerable time and money.

Numerous analysis are used to predict hate speech on social media websites. Semantic text and hate content are tracked by applying natural language processing (NLP) [7] and Artificial Intelligence (AI) [8] techniques. Any effort to develop an automated hate speech detection and prediction system must ensure that the method is scalable, reliable and sustainable due to the enormous volume of Internet content. The automated system proposed in this paper analyzes texts and classifies content into hate speech and non-harmful speech. During the hate speech recognition process, companies must flag unwanted messages [9].

This work proposes ways to use NLP to predict hate speech on social media websites. NLP was published by Alan Turing in the 1950s. The NLP process [10] in our model was hand-coded in the 1980s for use in automatic translation. This automatic speech translation system uses machine learning methods, which use a set of rules that enable computer analysis of a vast quantity of words that have system or human annotations. Integrating the machine learning method with an NLP approach has several advantages, including the capacity to automatically

detect text and reduce recognition times, by successful utilization of learning procedures, enable easier and more accurate decisions about text use and translation.

The NLP's primary tasks are syntax and semantics [11]. Syntax includes several tasks, such as grammar induction, lemmatization, morphological segmentation, speech tagging, parsing, sentence breaking, and stemming. Semantics includes the tasks of lexical analysis, entity recognition, language generation, language understanding, character recognition, sentiment analysis and so on. Various logical and statistical computational techniques are used for these syntax and semantic analyses [12]. The logical techniques utilize a set of rules for extracting words from sentences and mapping each word, with defined rules for recognizing particular languages. Statistical techniques extract patterns from huge volumes of language corpora. Although the NLP technique works perfectly, but it is hard process because language exists on several levels: sound, word, syllable, dialogue, and sentence. The NLP process must understand written and spoken words, as well as the rules that define the relationship between structure and meaning. This is why in this system combining the NLP process with the artificial intelligence techniques, such as bag of word extraction, word embedding, support vector machine, naïve Bayes (NB), and the random forest method.

The proposed methods were analyzed using 1500 samples to determine the effectiveness of incorporating machine learning techniques [12] with NLP. The automatic system was found to improve the recognition and prediction rate of hate speech on social media websites. In addition, this method was found to be more accurate at recognizing hate speech and more time-efficient than the traditional. This is because the killer natural language processing optimizing ensemble deep learning approach (KNLPEDNN) was utilized to examine Twitter comments and effectively predict hate and non-hate messages. The proposed method used hundreds of Tweets as data during the self-learning process; it also categorized comments from past data analysis, which successfully reduced the misclassification rate. The efficiency of this system was evaluated using various datasets, and the system was implemented using a MATLAB tool. The main contribution of this work is to:

- Improve hate speech recognition rate and reduce the misclassification rate.
- Reduce complexity while collecting data and extracting hate speech.
- Ensure minimum loss during Twitter data analysis.

The rest of this paper is organized as follows: Section two analyzes various NLP-based language recognition process. Section three introduces the proposed KNLPEDNN-based hate speech detection process. Section four examines the effectiveness of the KNLPEDNN-based system. Section five makes observations about the overall automatic hate speech recognition system.

2 Related work

The automatic system for recognizing hate speech on social media was developed in accordance with certain research, reviews, and methodologies. These methodologies resources, algorithm steps, text features were analyzed in order to produce a model

for identifying textual hate speech in different languages [13]. The hate speech detection system proposed here will help reduce illegal activities and communications social media websites. The efficiency of this system was evaluated using different language datasets to predict hate speech. This method for identifying hate speech will maximize the capacity of social media websites to meet their responsibilities to combat hate speech.

In order to detect hate speech in Twitter data, this system uses a recurrent neural network to collect Twitter data according to the sexism or racism information [14]. The collected details are processed by a network that examines frequent words and textual information to predict negative comments that are likely to derive from a post. During analysis, system collects 16,000 Tweets in order to examine the efficiency of our recurrent network-based detection process. The method effectively classifies the Twitter data into sexist or racist tweets and normal Tweets, so this system should improve the hate speech classification process.

The system examined offensive and hate speech on social media websites using N-gram, term frequency and inverse document frequency analyses, along with machine learning techniques [15]. The collected Twitter data, then hate posts were examined using n-gram analysis. They were then classified using a variety of machine learning approaches: vector machine, logistic regression, and Naïve Bayes. The proposed method successfully predicted features of incoming Tweets that could be used to classify Tweets into clean, hateful, and offensive language. The system introduced in this paper was able to predict toxic messages on social media websites with 95.6% accuracy; this was determined using MATLAB-based implementation results.

Performing massive analysis of this system by applying an ensemble approach to hate speech recognition on Indonesian language Twitter [16]. The method collected Indonesian Twitter data, and processed it using machine learning approaches, such as k-nearest neighbor, Naïve Bayes, support vector machine, random forest approach, and maximum entropy. The data was then processed using hard and soft ensemble voting, which successfully classified the Twitter data into hate speech and neutral speech. The system predicted hate speech with up to 84.7% accuracy, and the voting-based ensemble learning concept also reduced errors in the classification process.

Resolving classification problem by detecting hate and non-hate text on Twitter using NLP and neural network [17]. During analysis, collected long tail Twitter data. NLP techniques were used to predict and extract various text features from that data, which were fed into the deep learning neural network to successfully categorize Twitter text. The efficiency of this system was evaluated using F-score and accuracy, and it was found to effectively minimize hate speech. This paper introduces the integration of a transformed word embedding model with NLP technique, which can be used to predict text based on a large volume of data. A word2vec technique that identifies the context of particular words by analyzing their relationship to neighboring words according to the fuzzy concept linguistic principle [18]. This process must be continuously repeated to derive contextual information from a text.

Information extracted by applying NLP together with the support vector machine technique [19]. This method resolves the uneven margin problem by extracting

information using the passive learning process. The efficiency of the system was evaluated using multiple of datasets and job corpus. The proposed system was able to retrieve the requisite information in less time, with greater computational efficiency. It also introduces the character-aware NLP model [20], which can examine the characters present in a text. The system utilizes different neural networks alongside NLP, including convolution, recurrent and long short-term memory models to predict text according to user requests. The efficiency of the system was evaluated using experimental analysis and semantic information. The literature demonstrates that hate speech on social media websites must be removed using continuous tracking processes; however, the currently available automatic detection systems fail to identify certain complex textual features of hate speech, which reduces overall recognition accuracy.

The proposed recognition system uses optimized NLP integrated with KNLPEDNN, a machine learning technique, to predict hate speech. It examines collected Twitter data features in unique ways, to extract features that help accurately predict hate speech. In addition, the ensemble learning process minimizes weak classifiers, which improves the overall classification process and particularly minimizes misclassification.

3 KNLPEDNN-based hate speech detection

3.1 Twitter dataset collection

This process detects hate speech by integrating NLP with an ensemble deep learning approach. The automatic system requires large volume of Twitter data in order to learn how to track and identify hate speech. Sentence level hate speech was collected from *Stormfront*, a neo-Nazi website [21], and that dataset consists of 10,568 sentences. These sentences are ± 20.39 words in length; they primarily consist of attacks based on religion, sexual orientation, nationality, disability, ethnicity, gender, social class, and identity politics. The collected information enabled the machine learning algorithm to classify the dataset into hate and non-hate speech.

Sample hate speech was defined as, "white people are blessed by God and other people will go to Hell". In this example, the first part of the Tweet related only to white people, while the hate speech referred to people of color. A sample non-hate speech sentence was defined as, "The Gaana app help people successfully download songs in any language." In addition to the *Stormfront* database, used a Twitter database owned by a machine learning company, CrowdFlower [22], to examine hate, non-hate and offensive sentences. The CrowdFlower dataset consists of 24,783 Tweets in which 1430 hate Tweets, 19,190 offensive(hate) Tweets, and 4163 neither Tweets. These analysis showed that 5% of Tweets hate speech relevant various aspects of personal identity, 77% were offensive Tweets, and the remaining Tweets were unbalanced (confused tweets), so that portion of the dataset was omitted from the hate speech detection analysis. NLP with machine learning techniques were used to process the data and then to predict hate speech on Twitter. This process of analysis is detailed in the following section.

3.2 Twitter data processing using killer NLP

3.2.1 Data preprocessing

This hate speech detection system was created because of the impact hate speech about marginalized identity characteristics has on social media websites. Online hate speech is difficult to control, and efforts to filter hate content are complex. This paper introduces a method for using NLP with machine learning to categorize language as hateful, offensive, or neither.

During data collection, the killer application [23] is used to scrape various comments relevant to a particular issue on Twitter. The killer app is a programming language used to collect the hardware, software and consumers needed on a specific platform. The Twitter data collected in this way will always extraneous information; this increases the complexity of detecting hate speech on Twitter. Thus, NLP has to carry out several key tasks in order to improve the process of identifying hate speech:

1. Unwanted characters [24] are removed from Twitter data. Take, for example, this Tweet: http://hub.am/Sgsvt5 #Twitter tips-Hate it by @HubSpot". The Tweet includes several unwanted symbols that complicate the hate speech detection process, such as *URL*, */*, @ and #. The cleaning process involves analyzing the posted Tweet and removing unnecessary content, so after the URL and tag (i.e., "@username") are eliminated, the Tweet becomes:"#Twitter tips-Hate it by." Further analysis allows the removal of irrelevant content from the Tweet, which becomes "#Twitter tips Hate it". Last, Twitter hashtags are decomposed, so the final version of the Tweet is "Twitter tips Hate it."
2. After extraneous characters are removed, a stemming process [25] is applied to derive the root words in the Tweet. Stemming an effective NLP approach because successfully processes words by identifying their root or origin. During this process, a lookup table is maintained to manage incoming words related root word. NLP stemming identifies origin words by applying a suffix striping method. For example, if a word ends with *-s, -ing, -ed* or *-ly*, that suffix must be removed to identify the root word. In the example Tweet, the root of the word "tips" is identified as "tip".
3. NLP tokenization is the next step [26], in which Tweets are analyzed using the OpenNLP tool. The tokenization process involves using the *sentDetect()* function to detect the start and end boundaries of each sentence. After individual sentences are identified, tokenization breaks the sentence into smaller sentences. The *Token()* function is used to perform the tokenization process effectively. Then, parts of speech (PoS) [27] such as noun, verb, adjective, adverb, singular, plural and remaining PoS is detected using respective open NLP function. This helps determine negative meanings in the sentence. The process of identifying and segmenting sections of the sentence are repeated down to individual words.
4. Finally, a negation vector is generated using the lookup table. The table consists of root words and assorted stemmed words. From that, negative words are analyzed and some are assigned a -1 value. The remaining words are assigned a $+1$

value. This process is repeated for entire collected Twitter data in the automatic hate speech detection process.

The overall Twitter data preprocessing steps are represented in Fig. 1.

Figure 1, below, illustrates the overall Twiter data preprocessing steps by which incoming Tweets are examined using NLP, noise data is removed. The preprocessed data is fed into an automatic hate speech recognition process, called feature extraction, discussed in the next section.

3.2.2 Twitter data-feature extraction

Feature extraction is an important step in the automatic hate speech recognition process. The NLP techniques determines four different features of each Tweet [28]: semantic feature, sentiment-based feature, unigram, and pattern feature. These features are mostly relevant to the expression emphasized in the hate speech, how explicit or implicit it is, and emotional level associated with it. These features can then be used to predict hate content. Hate speech is identifiable in these features as Tweets containing only offensive words. NLP processing is applied to Tweets that are already cleaned, as described in Sect. 3.2.1.

Semantic features are the first aspect of the processed Tweets that are analyzed. This involves identifying interjections, capital words, punctuation. The sample semantic feature list is represented in Table 1.

The sematic features listed in Table 1 are derived only from the preprocessed tweets. After semantic features are analyzed, sentiment features [29] must be identified because they reflect the human emotional level associated with a particular Tweet. For this, Tweets are categorized as being negative, positive, or neutral. For each tweet, a score is estimated using positive words (P) amd negative words (N) to determine the sentiment of the Tweet. The ratio is computed as follows:

$$pn(t) = \frac{P - N}{P + N} \tag{1}$$

In Eq. (1), emotional word value is set as 0 when the Tweet does not have any emotional words. Various sentiment features, such as the quantity of positive and negative words, emoticons, and hash tags are computed to determine the polarity, the positive or negative level of the Tweet. Negative words, slang, and emoticons are assigned a value from -5 to -1 while positive words, slang, and emoticons have a value from 1 to 5. These values are used to predict hate speech on social media websites.

Next, unigram features are derived from a given Tweet according to parts of speech. A Tweet consists of two values, false and true, which means that the sentence is likely to have negative or positive comments. In addition, each word, *w,* in a Tweet is categorized as class C_i (neither, offensive and hate language). This enables the model to estimate the probability of a word appearing in each Tweet. Ni is denoted as the number of words that appeared in class *i*. The ratio is estimated as follows,

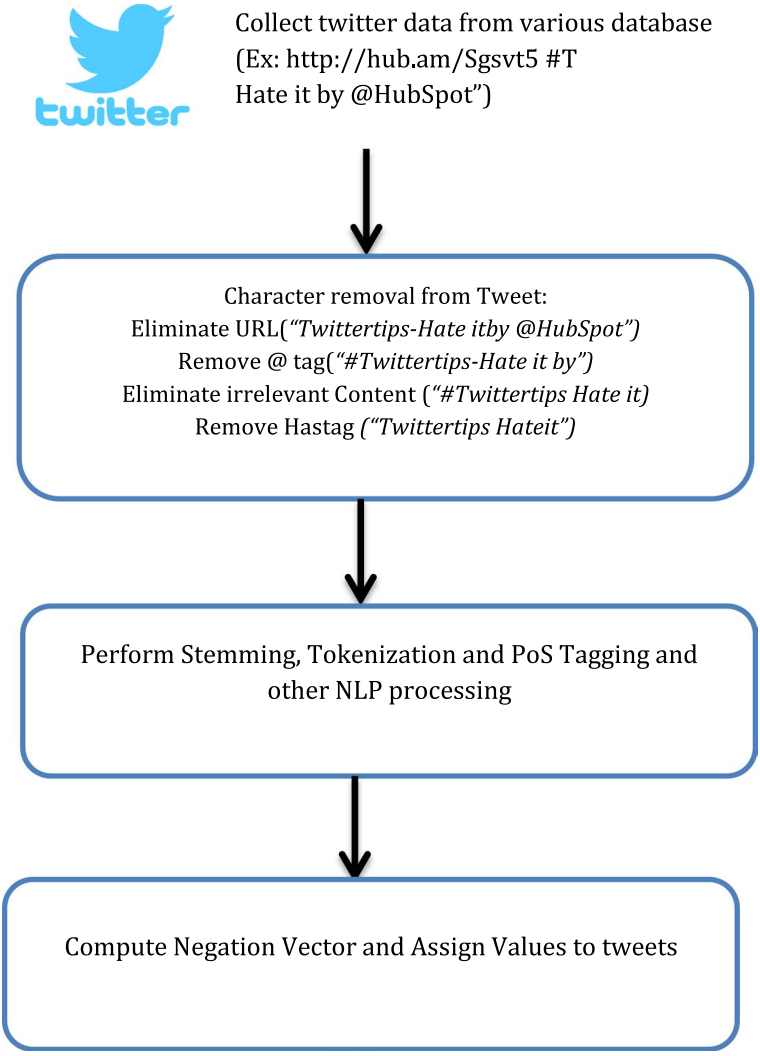

Fig. 1 NLP-based twitter data preprocessing steps

$$p12(w) = \frac{N1(w)}{N2(w)} \tag{2}$$

$$p13(w) = \frac{N1(w)}{N3(w)} \tag{3}$$

In Eq. (2), the denominator ratio is zero, then it is set as 2. This process repeated for every word in the class until the following condition is satisfied:

$$pij(w) \geq Th_u \tag{4}$$

Table 1 Semantic features

No	Semantic feature list	Example (*Twitter tip Hate it)*
1	Total words in the tweet	4
2	Number of question marks in tweet	Nil
3	Total number of comma in tweet	Nil
4	Number of exclamation in tweet	Nil
5	Full stop in tweet	1
6	Capital words in tweet	1-Hate
7	Total interjection in tweet	1-Hate
8	Total number of expressions in tweet	1-Hate

In Eq. (4), pij is the relation between two words in a class.

Th_u is the threshold value, defined as the ratio of class values. It set by default as 1.4.

Based on the above process, each class of word is examined and the value is set as true or false based on the ratio condition.

Finally, pattern features are derived from the training set using sentimental and non-sentimental features. Each word in a Tweet is evaluated in terms of parts of speech natural language processing: it belongs to the categories noun, adverb, adjective, and verb, and it also belongs to a sentimental or non-sentimental class. The length of the Tweeted word is also estimated because word length is fixed as 7. If the word is above that maximum length, different pattern features [30] are derived, or else that word is removed from the list. Then the ratio of class patterns is examined as follows:

$$p12(pattern) = \frac{N1(pattern)}{N2(pattern)} \tag{5}$$

$$p13(pattern) = \frac{N1(pattern)}{N3(pattern)} \tag{6}$$

In Eq. (5 and 6), the denominator is zero, and then it is set as 2. This process repeated for every word in the class until the following condition is satisfied:

$$pij(pattern) \geq Th_{pattern} \tag{7}$$

In Eq. (7), *pij* is the relation between two patterns in a class. $Th_{pattern}$ is the threshold value, defined as the ratio of class values. It is set by default as 1.4.

In this analysis, 1538 words were extracted. 1538 uniform features and 1873 patterns were derived. If a patterns is present in the Tweet, it is assigned the value 1; if the pattern is not present, the value is 0. In addition, the number of Tweets is n out of N; the optimized value is represented as, $\alpha * n/N$ ($\alpha = 0.1$), which is the default.

This NLP system for analyzing semantic, sentiment, unigram and pattern features of hate speech is further optimized by the application of machine learning techniques discussed in the following section.

3.3 Hate speech detection using an ensemble deep learning approach

The final step this model predicting hate speech [31] using the sentiment, semantic, unigram and pattern features derived through NLP analysis. Deep neural networks [32] can access additional hidden layers while processing those NLP-derived features, which enables the prediction of hate speech. The network consists of an input layer, more than three hidden layers, and an output layer that produces the output of a given Tweet. During the analysis process, the network utilizes the ensemble learning method to improve its overall prediction rate to minimize output variance. Each layer of the network trains the set of NLP-derived features according to the output of previous nodes. This is accomplished by using defined nodes created from aggregated Tweet features present in a previous layer. The utilization of output from one layer in another layer, called "feature hierarchy," increases prediction complexity. The deep learning process utilizes the high dimensionality of data which minimizes the complexity with the help of non-linear functions. Greater numbers of hidden layer improve processing of input-related output from previous layers and past analysis. A sample deep learning network structure is shown in Fig. 2.

Figure 2 depicts a sample deep neural network that consists of three hidden layers that help process input. During the output estimation process, network weight value must be optimized to minimize error rate. In this work, gradient decent optimization function was used for this purpose. The gradient decent [33] is also the slope used to predict the relationship between two features' weights in networks that completely minimize feature variance. The gradient descent process recognizes the relationship between weight and network error using the chain rule estimation process.

$$\frac{dError}{dweight} = \frac{dError}{deactivation} * \frac{deactivation}{dweight} \tag{8}$$

Equation (8) defines how network weight is affected by network function, which is measured in terms of network error. Thus, the gradient optimization

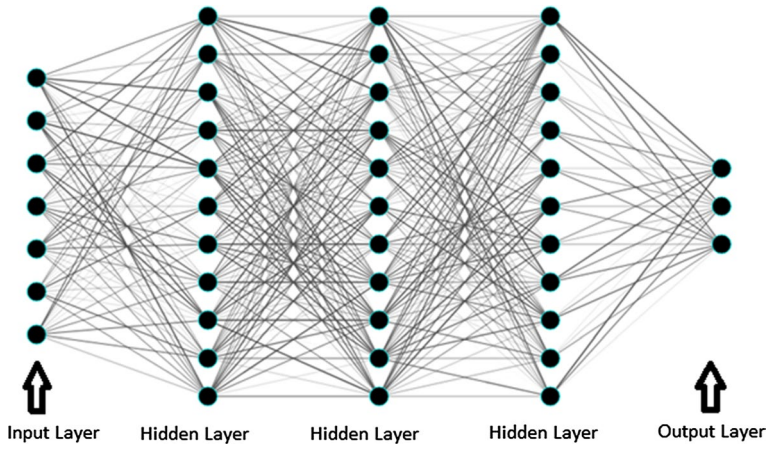

Fig. 2 Sample deep neural network structure

function minimizes network deviation. Further, the network weak learning [34] are reduced by the ensemble learning process, which maximizes the overall hate speech detection process. The derived NLP features are represented in terms of input and output $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, because each input has a particular labeled output value. The NLP features are fed into the network with a particular weight value, denoted as $\{w_1, w_2, \ldots w_n\}$. After collecting the features and weighing their value, the weaker learner is expressed as follows:

$$D_{(m-1)}(x_i) = \propto_1 w_1(x_i) + \cdots \propto_{m-1} w_{m-1}(x_i) \tag{9}$$

The network also denotes particular exponential loss in Tweet features as:

$$E = \sum_{i=1}^{N} e^{-y_i D_m(x_i)} \tag{10}$$

Based on the loss function, network weight value is updated continuously to minimize the error rate effectively, as shown in Eq. (11):

$$W_{i,t+1} = W_{i,t} e^{-y_i \propto_t h_t(x_i)} \tag{11}$$

The $\propto_t$ is calculated as:

$$\propto_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \tag{12}$$

$$\epsilon_t = \sum_{i=1}^{n} W_{i,t} \quad //\text{misclassification point} \tag{13}$$

After optimizing network weight, the activation function is applied to calculate the network output because the last layer of the network plays an important role in predicting output. NLP Tweet features are processed in each layer; unlabeled hate speech features are also examined using past history because the deep network extracts these features without human input. The estimated output, calculated from the derived features, is compared to the training set features to determine the exact output. At the time of output computation process, a logistic regression function is used to compute the output. This is estimated as follows:

$$F(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

Based on the above process, the given NLP tweet features are analyzed in order to produce three outputs: neither (0), offensive (−1), and hate language (1). This process is performed for every new incoming NLP-based derived Tweet. The algorithm, based on the above equations, is depicted as follows (Table 2).

The algorithm steps detail how hate speech on Twitter is recognized. The efficiency of the system is evaluated using experimental results.

Table 2 Algorithm for hate speech recognition

Step 1: Collect Twitter data from crowd flower and storm front dataset
Step 2: Remove characters, URL, hash tag, irrelevant content from tweet
Step 3: Extract root word from tweet
Step 4: Extract parts of speech from tweet by segmenting sentence level tweet into words
Step 5: Generate the look up table to maintain the PoS and other information about tweet
Step 6: Extract semantic features such as, words in tweet, question marks, comma, exclamation, full stop, capital word, interjection and expressions from tweet
Step 7: Derive sentimental features, unigram features and pattern features from tweet using Eqs. (1 to 7)
Step 8: Assign values to features depending on the negation words in tweet
Step 9: Extracted features are fed into the deep classifiers
Step 10: Process the input, and weight value of the network is analyzed by gradient decent process which is computed using Eq. 8
Step 11: Minimize the weak features using ensemble classifiers that is represented in Eq. (9 to 13)
Step 12: Update the network weight value according to the step 11
Step 13: Compute the output using logistic regression process defined in Eq. 14
Step 14: Repeat this process from step 2 to step 13 to predict the hate, offensive and neither language in Twitter

4 Results and analysis

The efficiency of the KNLPEDNN-based hate speech detection system is evaluated in this section. At the time of implementation, NLP integrated with machine learning library functions are implemented using Python. As discussed in Sect. 3.1, Twitter datasets including Stormfront and CrowdFlower, were used to demonstrate the capacity of this method of detecting hate speech [35] through the application of several steps of NLP processing and machine learning procedure. The datasets consist of thousands of Tweets that need to be examined continuously at the sentence and word level so that social media websites can eliminate negative communications in particular situations. 80% of the data collected from Twitter was utilized as training model for this purpose. The keras library function and ensemble optimized deep learning approach were used. The remaining 20% of Twitter data was used as the testing model, in which the Tensor flow backend was used to develop the hate speech detection system.

The efficiency of this, the implementation stage of a proposed automatic hate speech detection model for Twitter [36], is evaluated using several performance measures: accuracy, f-score, precision, recall metrics, and the Receiver Operating Characteristics (ROC) metric. First, the system variance or deviation from expected to predict value must be identified using loss functions such as mean square error rate, cross entropy loss, and likelihood loss. The defined loss function helps to predict how effectively KNLPEDNN works to predict hate speech in the Twitter dataset. The loss function examines the difference between predicted value $\hat{y}$ and labeled value y. The obtained loss function values of KNLPEDNN with traditional classifiers Transformed Word Embedding Model with Multi-Layer Perceptron Network (TWEM-MLP) [37], NLP support vector machine (NLP-SVM) [38], convolution-gru based deep neural network (CGDNN) [39] and

character-aware neural language models with neural network (CANLNN) [40]. The mean square error rate [41] is computed as:

$$Mean\ square\ error\ rate = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{15}$$

In Eq. (15), N is the quantity of data relative to predicted value $\hat{y}$ and labeled value y. Following this, cross entropy loss [42] is estimated as:

$$Cross\ entropy\ loss = -\sum_{c=1}^{M} y_{o,c} log\left(p_{o,c}\right) \tag{16}$$

In Eq. (16), M denotes classes and related features, such as neither, hate, and offensive speech.

O is the observed value of particular class-related feature. P is the prediction probability value relevant to O.

Log, is the logarithmic function, and y is the output value of a specific class, which has the binary values 0 and 1.

In addition, likelihood loss [43] is computed in order to estimate the accuracy of KNLPEDNN at identifying hate speech on Twitter. It is estimated as follows:

$$L = -\frac{1}{n}\sum_{i=1}^{n} log\left(\hat{y}^{(i)}\right) \tag{17}$$

In Eq. (17), n is number of classes and output is denoted as y. The methods loss function values are defined in Table 3.

Table 3 compares the loss function value of several traditional automatic hate speech detection system approaches with KNLPEDNN. It shows that the KNLPEDNN approach achieves minimum loss function values in relation to the other methods, which means minimal deviation between predicted and estimated output values. That smaller deviation is achieved through the successful utilization of semantic, sentiment, unigram and pattern NLP features. The fact that the learning classifier analyzes extracted features using past layer information also minimizes the deviations. Figure 3 is a graphic representation of this loss function.

Figure 3 shows different loss functions, specifically mean square error rate, cross entropy loss and likelihood loss value of various classifier. The figure clearly shows that KNLPEDNN achieves the lowest loss function (MSE 0.019, CEL 0.015 and LL 0.0238) compared to the other methods, such as TWEM-MLP (MSE 0.26, CEL 0.218 and LL 0.324), NLP-SVM (MSE 0.327, CEL 0.352 and LL 0.376), (MSE 0.432, CEL 0.467 and LL 0.472) and CANLNN (MSE 0.532, CEL 0.563 and LL 0.587). The reduced loss function demonstrates that the KNLPEDNN method detects and recognizes hate speech with greater efficiency that can be measured using precision and recall metrics. The precision value measures how often KNLPEDNN can correctly identify hate speech in different quantities of Twitter data. Precision [44] is calculated as follows:

Table 3 Loss function

Methods	Mean square error rate (MSE)	Cross entropy loss (CEL)	Likelihood loss (LL)
KNLPEDNN	0.019	0.015	0.0238
TWEM-MLP	0.26	0.218	0.324
NLP-SVM	0.327	0.352	0.376
CGDNN	0.432	0.467	0.472
CANLNN	0.532	0.563	0.587

$$Precision = \frac{True\ positive}{True\ positive + false\ positive} \tag{18}$$

In Eq. (18), a true positive refers to how often the KNLPEDNN method correctly identified or predicted negative speech, while a false positive refers to the number of incorrect predictions in the given data. The obtained precision value for a variety of methods is depicted on Table 4.

Table 4 compares the precision values of KNLPEDNN with the traditional hate speech recognition methods, TWEM-MLP, NLP-SVM, CGDNN and CANLNN with regard to two distinct Twitter datasets. KNLPEDNN attains high precision value because the method is better at identifying token, POS and other negation vectors relative to words and sentences in the Twitter data. In addition, the huge volume of previous analyzed Twitter data is incorporated into the optimization function, which improves the recognition rate. Figure 4 is a graphic representation of Table 4.

Figure 4 represents the precision value of KNLPEDNN with regard to two Twitter datasets, Stormfront and CrowdFlower. Figure 4 demonstrates that KNLPEDNN

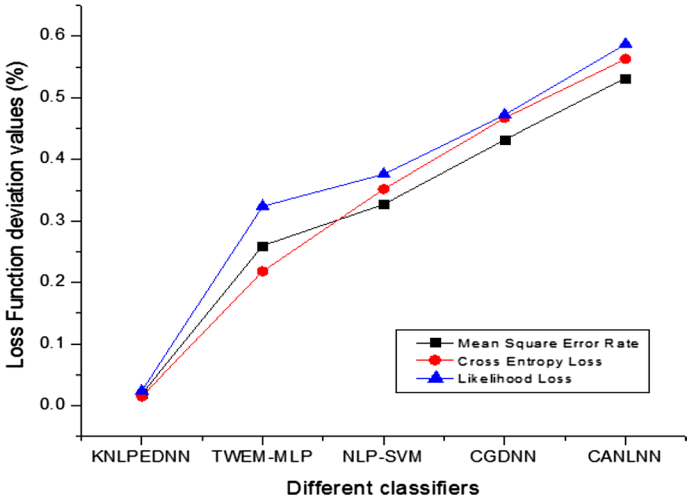

Fig. 3 Loss function

Table 4 Precision

Methods	Number of tweets									
	Storm front dataset					Crowd-flower dataset				
	500	1000	1500	2000	2500	500	1000	1500	2000	2500
KNLPEDNN	98.945	98.42	98.84	98.41	98.52	98.79	98.76	98.64	98.69	98.724
TWEM-MLP	96.57	96.89	96.83	97.01	97.21	96.90	96.863	97.01	97.03	96.254
NLP-SVM	95.35	95.7	95.98	96.01	96.32	95.83	95.76	96.36	95.87	95.76
CGDNN	94.27	94.67	94.89	95.03	95.23	94.98	95.02	95.43	94.72	94.87
CANLNN	93.5	93.56	94.02	94.21	93.89	93.93	93.76	94.04	94.21	93.87

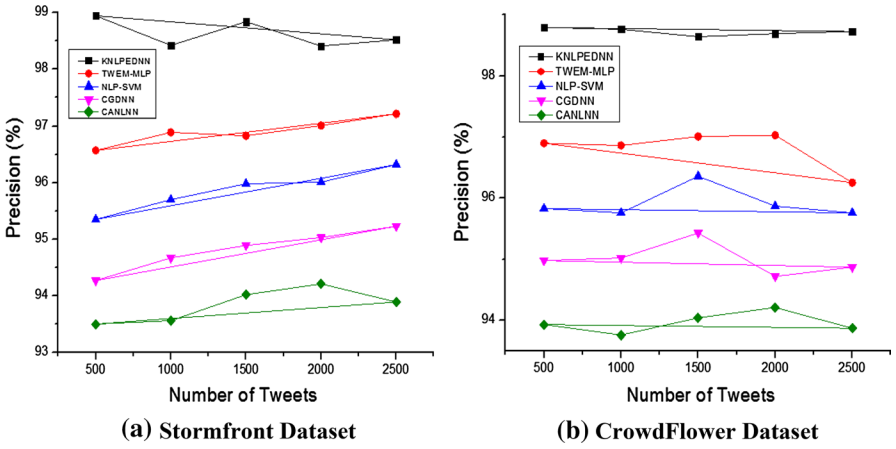

Fig. 4 Precision value

has the highest precision value with regard to Tweet prediction accuracy (Stormfront 98.62%, CrowdFlower 98.73%) compared to TWEM-MLP (Stormfront 96.90%, CrowdFlower 96.82%), NLP-SVM (Stormfront 95.87%, CrowdFlower 95.90%), CGDNN (Stormfront 94.81%, CrowdFlower 95%) and CANLNN (Stormfront 93.83%, CrowdFlower 93.95%).

In addition to precision, efficiency of recall is used to predict how effectively the KNLPEDNN method identifies the positive portion of Tweets. Recall [45] is estimated as follows:

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \tag{19}$$

In Eq. (19), true positive is defined in Eq. (18). True negative refers to how often the KNLPEDNN method correctly applied the negative class, and false negative refers to how often the model incorrectly recognized the negative class. The obtained recall value is shown on Table 5.

Table 5 Recall

Methods	Number of tweets									
	Storm-front dataset					Crowd-flower dataset				
	500	1000	1500	2000	2500	500	1000	1500	2000	2500
KNLPEDNN	97.87	97.63	97.83	97.98	97.9	97.86	97.92	98.1	97.96	98.03
TWEM-MLP	96.35	96.73	96.33	96.9	97.02	96.98	96.62	97.01	97.21	96.98
NLP-SVM	95.25	95.7	95.9	96.01	96.23	95.90	95.34	96.03	96.45	95.98
CGDNN	94.9	94.82	95.03	95.31	95.2	94.80	94.86	95.02	95.45	95.34
CANLNN	93.89	93.9	94.02	94.23	94.6	93.82	93.78	94.24	94.52	95.24

Table 5 compares the recall value of KNLPEDNN with traditional hate speech recognition classifiers, TWEM-MLP, NLP-SVM, CGDNN and CANLNN for two discrete Twitter datasets. The KNLPEDNN method has maximum recall value because the approach recognizes hate speech using the defined negation vector and PoS words. The recognized words improve the ensemble learner process by removing weak classifiers. Figure 5 is a graphic representation of Table 5.

Figure 5 demonstrates the recall value of the KNLPEDNN-based hate speech recognition process on two Twitter datasets, Stormfront and CrowdFlower. Figure 5 clearly shows that KNLPEDNN has a higher recall value (Stormfront 97.84%, CrowdFlower 97.97%) compared to other methods; TWEM-MLP (Stormfront 96.66%, CrowdFlower 96.95%), NLP-SVM (Stormfront 95.818%, CrowdFlower 95.94%), CGDNN (Stormfront 95.09%, CrowdFlower 95%) and CANLNN (Stormfront 94.12%, CrowdFlower 94.32%). The accuracy of the hate speech recognition process is evaluated with regard to predicting positive classes using F-score [46], which is computed as:

$$F{-}score = 2 \cdot \frac{precision * recall}{precision + recall} \tag{20}$$

Based on the Eq. (20), the F-score value is derived by using Eq. (18 and 19). The obtained value is shown on Table 6.

The successful derivation of NLP features from every Tweet enables the learning network to analyze the derived features using every node. The extracted features are also examined using several hidden layers that reduce the misclassification rate. In addition, the ensemble learning process improves the overall hate speech recognition rate, which is depicted on Table 6. Figure 6 is a graphic depiction of Table 6.

Figure 6 shows that KNLPEDNN has the highest hate speech recognition accuracy in both datasets (Stormfront 98.23% CrowdFlower 98.35%) compared to other methods, such as TWEM-MLP (Stormfront 96.74%, CrowdFlower 96.88%), NLP-SVM (Stormfront 95.85%, CrowdFlower 95.92%), CGDNN (Stormfront 94.93%, CrowdFlower 95.04%) and CANLNN (Stormfront 93.98%, CrowdFlower 94.14%). The KNLPEDNN method ensures high accuracy with minimum

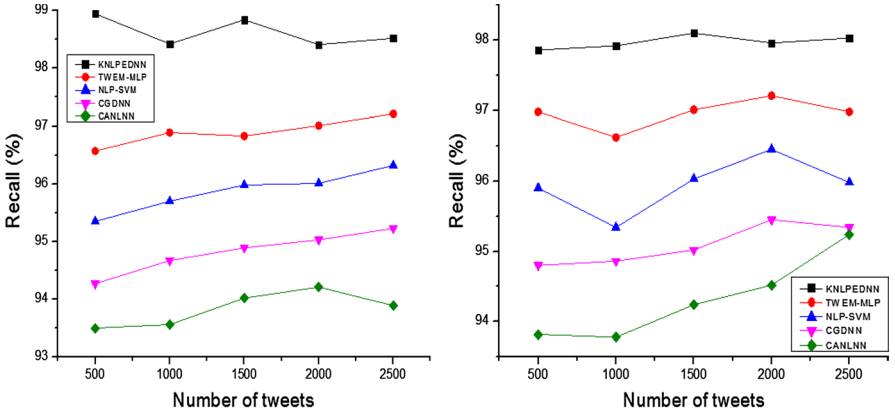

Fig. 5 Recall value

Table 6 F-Scores

Methods	Number of tweets									
	Storm-front dataset					Crowd-flower dataset				
	500	1000	1500	2000	2500	500	1000	1500	2000	2500
KNLPEDNN	98.4075	98.025	98.335	98.195	98.21	98.325	98.34	98.37	98.35	98.37
TWEM-MLP	96.46	96.81	96.58	96.955	97.115	96.94	96.741	97.01	97.12	96.61
NLP-SVM	95.3	95.7	95.94	96.01	96.275	95.865	95.55	96.195	96.16	95.87
CGDNN	94.585	94.745	94.96	95.17	95.215	94.89	94.94	95.225	95.085	95.105
CANLNN	93.695	93.73	94.02	94.22	94.245	93.875	93.77	94.14	94.365	94.555

loss function values, so the overall prediction rate must be high compared to other techniques. The obtained result is shown on Table 7.

Table 7 shows that that KNLPEDNN successfully analyzed the user Tweets in order to accurately predict hate speech, which was demonstrated in both testing and training process. Figure 7 is a graphic representation of Table 7.

The results and analysis section examines the capacity of the KNLPEDNN model to predict hate speech on Twitter. Figure 7 shows that the KNLPEDNN approach achieves the highest accuracy, 98.71%, which favorably compared to other methods such as TWEM-MLP (97.06%), NLP-SVM (95.61%), CGDNN (95.04%) and CANLNN (94.3%). This is due to the effective utilization of ensemble learning, because the activation function improves overall testing.

In addition to the above analysis, receiver operating characteristic curve (ROC) is used to evaluate the accuracy of the KNLPEDNN classification model. The ROC curve [47] evaluation includes the true positive rate or recall value and false positive rate, which is computed as follows:

$$False\ positive\ rate = \frac{False\ positive}{False\ positive + True\ negative} \tag{21}$$

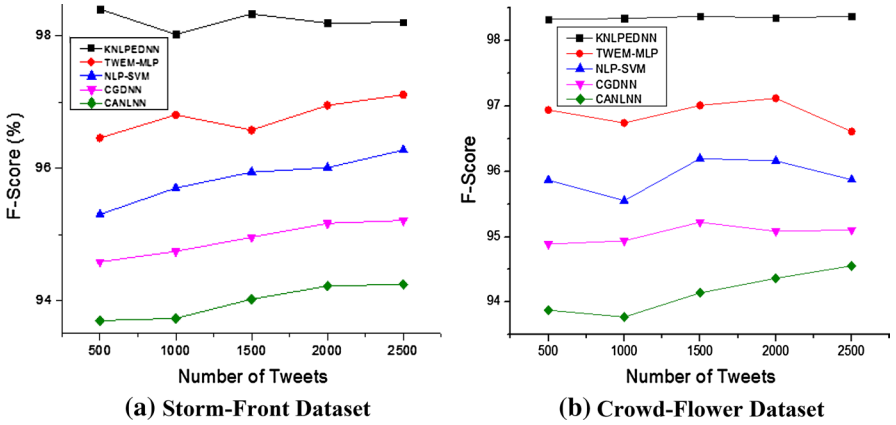

Fig. 6 F-scores

Table 7 Overall accuracy

Methods	Automatic hate speech recognition overall accuracy (%)		
	Training accuracy	Testing accuracy	Overall accuracy
KNLPEDNN	98.45	98.97	98.71
TWEM-MLP	96.89	97.23	97.06
NLP-SVM	95.25	95.98	95.615
CGDNN	94.87	95.21	95.04
CANLNN	93.98	94.62	94.3

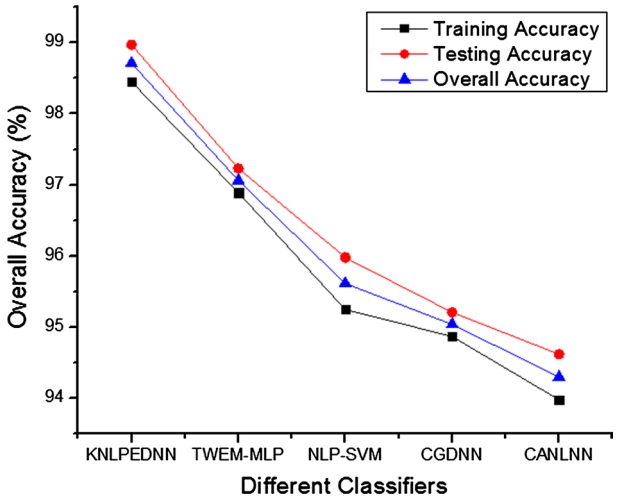

Fig. 7 Overall accuracy of hate speech recognition

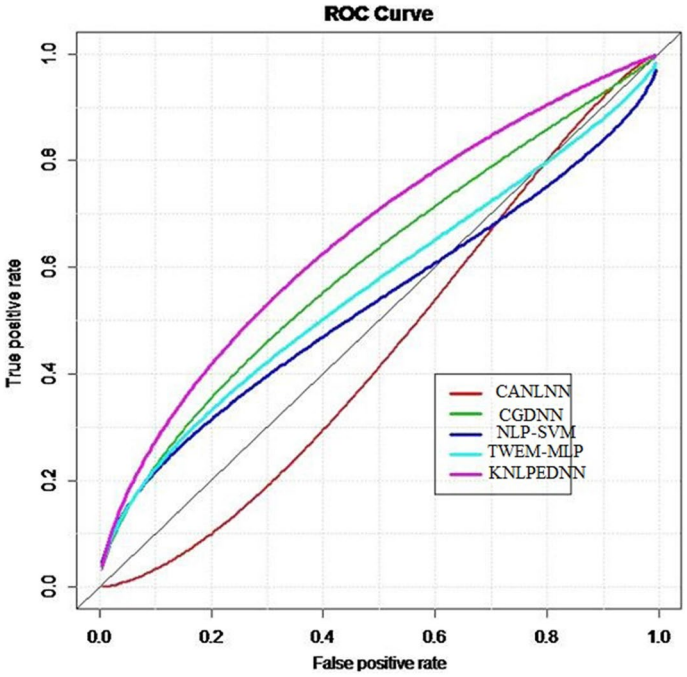

Fig. 8 ROC curve

Figure 8 is a graphic representation of the obtained ROC value. It clearly shows that the KNLPEDNN method was most effective at classifying hate speech on Twitter.

5 Conclusion

The KNLPEDNN-based hate speech recognition system is used to analyze Twitter data. Initially, tweets are collected from storm front and crowd flower dataset. The collected data are processed using an NLP approach. Data characters, hashtags, user information and other unwanted details are removed using the NLP tokenization process. The system analyzes Tweets in terms of sentence and words, and then it derives NLP features, which are called semantic, sentiment, unigram and pattern features. Vectors are computed from the derived features and values are assigned according to the patterns. After that, the features are processed using an ensemble deep learning classifier to predict whether responding Tweets will be classifiable as hate speech, offensive speech, or neither using an optimized function and weight updating process.

Finally, the system is implemented and evaluated using Python. The discussed procedure is applied to the Stormfront and Crowd-Flower Twitter datasets. It is demonstrated that the proposed automatic system ensures minimum loss function (MSE-0.019, CEL-0.015 and LL-0.0238) and maximum prediction accuracy (98.71%).

In future, this research would benefit from applying NLP processing with machine learning techniques to recognizing hate speech in audio and video formats.

Acknowledgements The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group No. RG-1439-088.

References

1. Xiang G, Fan B, Wang L, Hong J, Rose C (2012) Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM international conference on information and knowledge management. ACM, pp 1980–1984
2. Del Vigna F, Cimino A, Dell'Orletta F, Petrocchi M, Tesconi M (2017) Hate me, hate me not: hate speech detection on Facebook. In: Proceedings of the first Italian conference on cybersecurity (ITASEC17), Venice, Italy
3. 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
4. Watanabe H, Bouazizi M, Ohtsuki T (2018) Hate speech on twitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access 6:13825–13835
5. Bouazizi M, Ohtsuki TO (2016) A pattern-based approach for sarcasm detection on twitter. IEEE Access 4:5477–5488
6. Facebook, Google and Twitter agree German Hate Speech Deal. Website. http://www.bbc.com/news/world-europe-35105003. Accessed 26 Mar 2019
7. AlFarraj O, AlZubi A, Tolba A (2018) Optimized feature selection algorithm based on fireflies with gravitational ant colony algorithm for big data predictive analytics. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3612-0
8. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057
9. Chen Y, Zhou Y, Zhu S, Xu H (2012) Detecting offensive language in social media to protect adolescent online safety. In: 2012 international conference on privacy, security, risk and trust and 2012 international conference on social computing. IEEE, pp 71–80
10. Xia F, Liaqat HB, Ahmed AM, Liu L, Ma J, Huang R, Tolba A (2016) User popularity-based packet scheduling for congestion control in ad-hoc social networks. J Comput Syst Sci 82(1):93–112
11. Li J, Ning Z, Jedari B, Xia F, Lee I, Tolba A (2016) Geo-social distance-based data dissemination for socially aware networking. IEEE Access 4:1444–1453
12. Rahim A, Qiu T, Ning Z, Wang J, Ullah N, Tolba A, Xia F (2019) Social acquaintance based routing in vehicular social networks. Future Gen Comput Syst 93:751–760
13. Fortuna P, Nunes S (2018) A survey on automatic detection of hate speech in text. ACM Comput Surv (CSUR) 51(4):85
14. Pitsilis GK, Ramampiaro H, Langseth H (2018) Effective hate-speech detection in Twitter data using recurrent neural networks. Appl Intell 48(12):4730–4742
15. Gaydhani A, Doma V, Kendre S, Bhagwat L (2018) Detecting hate speech and offensive language on twitter using machine learning: an N-gram and TFIDF based approach. arXiv preprint arXiv:1809.08651
16. Fauzi MA, Yuniarti A (2018) Ensemble method for indonesian twitter hate speech detection. Indones. J Electr Eng Comput Sci 11(1):294–299
17. Zhang Z, Luo L (2018) Hate speech detection: A solved problem? The challenging case of long tail on Twitter. Semantic Web, (Preprint), pp 1–21
18. Chang CY, Lee SJ, Lai CC (2017) Sighted word2vec based on the distance of words. In: 2017 international conference on machine learning and cybernetics (ICMLC). IEEE, vol 2, pp 563–568

19. Alarifi A, Tolba A, Al-Makhadmeh Z, Said W (2018) A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. J Supercomput. https://doi.org/10.1007/s11227-018-2398-2
20. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Thirtieth AAAI conference on artificial intelligence
21. Caren N, Jowers K, Gaby S (2012) A social movement online community: stormfront and the white nationalist movement. In: Earl J, Rohlinger DA (eds) Media, movements, and political change (research in social movements, conflicts and change, volume 33). Emerald Group Publishing Limited, Bingley, pp 163–193
22. https://data.world/crowdflower/hate-speech-identification. Accessed 10 June 2019
23. Bergin TJ (2006) The origins of word processing software for personal computers: 1976–1985. IEEE Ann Hist Comput 28(4):32–47
24. Wong KF, Li W, Xu R, Zhang ZS (2009) Introduction to Chinese natural language processing. Synth Lect Hum Lang Technol 2(1):1–148
25. Gupta V (2014) Automatic stemming of words for Punjabi language. In: Thampi SM, Gelbukh A, Mukhopadhyay J (eds) Advances in signal processing and intelligent recognition systems. Springer, Cham, pp 73–84
26. Fares M, Oepen S, Zhang Y (2013) Machine learning for high-quality tokenization replicating variable tokenization schemes. In: International conference on intelligent text processing and computational linguistics. Springer, Berlin, Heidelberg, pp 231–244
27. Domínguez MA, Infante-Lopez G (2008) Searching for part of speech tags that improve parsing models. In: International conference on natural language processing. Springer, Berlin, Heidelberg, pp 126–137
28. Rahim A, Ma K, Zhao W, Tolba A, Al-Makhadmeh Z, Xia F (2018) Cooperative data forwarding based on crowdsourcing in vehicular social networks. Pervasive Mob Comput 51:43–55
29. Nicholls C, Song F (2010) Comparison of feature selection methods for sentiment analysis. In: Canadian conference on artificial intelligence. Springer, Berlin, Heidelberg, pp 286–289
30. Razavi AH, Inkpen D, Uritsky S, Matwin S (2010) Offensive language detection using multi-level classification. In: Canadian conference on artificial intelligence. Springer, Berlin, Heidelberg, pp 16–27
31. Chen Y, Zhou Y, Zhu S, Xu H (2012) Detecting offensive language in social media to protect adolescent online safety. In: 2012 international conference on privacy, security, risk and trust and 2012 international confernece on social computing. IEEE, pp 71–80
32. Jedari B, Xia F, Chen H, Das SK, Tolba A, Zafer AM (2019) A social-based watchdog system to detect selfish nodes in opportunistic mobile networks. Future Gen Comput Syst 92:777–788
33. Gomathi P, Baskar S, Shakeel PM, Dhulipala VS (2019) Identifying brain abnormalities from electroencephalogram using evolutionary gravitational neocognitron neural network. Multimedia Tools Appl. https://doi.org/10.1007/s11042-019-7301-5
34. Shakeel PM, Tolba A, Al-Makhadmeh Z, Al-Makhadmeh M, Musa J (2019) Automatic detection of lung cancer from biomedical data set using discrete AdaBoost optimized ensemble learning generalized neural networks. Neural Comput Appl. https://doi.org/10.1007/s00521-018-03972-2
35. Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. In: Eleventh international AAAI conference on web and social media
36. Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets. In: Proceedings of the 26th international conference on World Wide Web companion, pp 759–760
37. Yao Z, Sun Y, Ding W, Rao N, Xiong H (2018) Dynamic word embeddings for evolving semantic discovery. In: Proceedings of the eleventh ACM international conference on web search and data mining, pp 673–681
38. Hong G (2005) Relation extraction using support vector machine. In: International conference on natural language processing. Springer, Berlin, Heidelberg, pp 366–377
39. Zhang Z, Robinson D, Tepper J (2018) Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In: European semantic web conference. Springer, Cham, pp 745–760
40. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Thirtieth AAAI conference on artificial intelligence

41. Wackerly D, Mendenhall W, Scheaffer RL (2008) Mathematical statistics with applications, 7th edn. Thomson Higher Education, Belmont. ISBN 978-0-495-38508-0
42. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
43. Mikolov T, Deoras A, Kombrink S, Burget L, Černocký J (2011) Empirical evaluation and combination of advanced language modeling techniques. In: Twelfth annual conference of the international speech communication association
44. Powers DM (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. J Mach Learn Technol 2(1):37–63
45. Muhammed Shafi P, Selvakumar S, Mohamed Shakeel P (2018) An efficient optimal fuzzy C means (OFCM) algorithm with particle swarm optimization (PSO) to analyze and predict crime data. J Adv Res Dyn Control Syst 10(06):699–707
46. Shakeel PM, Manogaran G (2018) Prostate cancer classification from prostate biomedical data using ant rough set algorithm with radial trained extreme learning neural network. Health Technol. https://doi.org/10.1007/s12553-018-0279-6
47. Powers DM (2012) ROC-ConCert: ROC-based measurement of consistency and certainty. In: 2012 Spring congress on engineering and technology. IEEE, pp 1–4