Penguin Rider Optimization Algorithm-Based Deep Recurrent Neural Network for Sentiment Classification of Political Twitter Data

Vegi Harendranath, Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University (Deemed), India*
Sireesha Rodda, Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University (Deemed), India

ABSTRACT
This paper proposes an effective and optimal sentiment classification method named penguin rider optimization algorithm-based deep recurrent neural network (PeROA-based deep RNN) to perform sentiment classification using political reviews. However, the proposed PeROA is developed by incorporating the penguins search optimization algorithm (PeSOA) with the rider optimization algorithm (ROA). The sentiment classification process is progressed using the deep RNN classifier, which in turn generates the optimal solution based on the fitness measure. Accordingly, the function with the minimal error value is accepted as the best solution. The sentiment-based features enable the classifier to perform better classification result with respect to the sentiment tweets. However, the proposed PeROA-based deep RNN obtained better performance using the metrics, like accuracy, sensitivity, specificity, recall, F-measure, thread score, NPV, FPR, FNR and FDR, with the values of 92.030%, 92.030%, 92.235%, 92.030%, 92.030%, 92.030%, 92.030%, 3.105%, 3.11%, and 3.105%, respectively.

KEYWORDS
Penguin Search Optimization Algorithm (PeSOA), Political Reviews, Rider Optimization Algorithm (ROA), Sentiment Classification, Twitter Data

1. INTRODUCTION
In the recent decades, social media is widely expanded to deal with the social lives. However, the social media platform, like twitter and Facebook plays a major role in the way that how people interact with the politicians and world. The social media gained a significant economic role in the business activities, as the business strategy uses the social media as an integral part in the marketing area by considering the advantage of direction contact with the users (Mattila and Salman, 2018). E-commerce platforms, web forums, product reviews, stock market selling and buying, social media platform, feedbacks, and online blogs have changed the public data as the source (Dingli, et al., 2015a; Joshi and Simon, 2018). However, the public share the things that are happening round them, important
reviews, opinions, and the geographic location. All the individual public is termed as social sensors in such a way that this information is grouped as subjective information. For instance, subjective information is termed as the collection of opinions and sentiments (Sakaki, et al., 2010; Guerrero, et al., 2015; Joshi and Simon, 2018). Millions of people use the social media around the world in order to connect to their coworkers, family members, and friends through mobile phones and computers. Various services are introduced in the social media to achieve target audience by offering some means of communication (Fu, et al., 2019; Yang, et al., 2019; Dingli, et al., 2015b). However, social media is the platform, where the public can socially interact with others by using web based technology for quickly disseminating the details and information with the wide range of users.

According to Wikipedia, it is stated that social media is the internet-based resources used to discuss and share the details with the public. Social networking websites that are related to the community specifies the opinions and comments of the areas and the parties engaged with them. However, any website that allows the user to discuss the review, motivates the connection, group development, and material can be categorized as social media (Ma, et al., 2019; Zhang, et al., 2019; Si, 2016). However, communication in the political review is the process of posting the technology, media and the information in the service. In the social revenue, people can share the contents with others and allows the sharing of political data. With web 2.0 technology, the public can greatly offer political comments and information to other users with reduced professional communications (Song, et al., 2019; Zhang, et al., 2019; Bode, 2016). However, the communication that is related to the politics and the press are historically defined using the social conditions, which specify the communication processes and the audiences to transfer the messages with the effect of communication process. The focus of democratic society occupies more in the area of political reviews such that the concepts and core theories gained the assumptions regarding the well defined public spheres, where the communication is passed from the legitimate user in order to affect the actions and the opinions of citizens through the organization of press (Bennett and Pfetsch, 2018). In the recent decades, number of students suggested that there exists positive connection between the media system over strong service media for delivering the political knowledge and political information (Huang and Carley, 2019; Xue, et al., 2020; Stromback, 2017).

The users in the social media share large amount of information and news in the online platforms with the political key moments. With that instance, the political actor as well as the internet subcultures are greatly adapts the computational resources for disseminating the information to the citizens. However, the news reaches the user by browsing the social media (Glowacki, et al., 2018). However, the potentiality of the social media appears more promising in the view of political context that make to perform more democracy and participation. In (Nio and Murakami, 2018), Apache spark and the MapReduce framework are developed to exploit the comments from tweets by considering the large scale dataset in sentiment analysis. In (Kanavos, et al., 2017), the sentiment analysis model is designed for identifying the social communities based on the influential impact and is implemented by considering the metric value for each public emotional post.

This research focuses to develop the sentiment classification approach using the proposed PeROA-based Deep RNN for classifying the sentiment tweets as either positive or negative tweet using the sentiment reviews. At first, the input political review data is passed to the pre-processing stage, where the stop words are removed and the stemming process is used to remove the noise of data and to reduce the dimensionality of sentiment tweets. The pre-processed data is fed to the feature extraction module, where the sentiment based features, like hashtags, elongated words, multiple exclamation marks, absence or presence of greeting words, multiple question marks, and number of sentences is effectively extracted from the sentiment reviews. After extracted the features, it is highly required to select the essential features so that the classification performance can be increased. The feature selected process is achieved by the mutual information. Finally, the Deep RNN classifier is used to classify the tweets such that the Deep RNN is trained by the proposed PeROA.

The major contribution of this research is explained as follows:
• **Proposed PeROA-based Deep RNN:** An effective sentiment classification approach is performed using the proposed PeROA-based Deep RNN. Accordingly, the optimal features are uniquely selected using mutual information. The classifier effectively performs well with the selected features, which in turn increases the performance of classification through the evaluation of fitness measure. Moreover, the weight update process is performed by the proposed optimization in such a way that it reflects the performance of classification with minimal fitness value.

The paper is organized as follows: section 2 describes the review of existing classification methods. Section 3 elaborates the proposed PeROA-based Deep RNN for sentiment classification using the political reviews. Section 4 explains the results and discussion of the proposed classification approach, and finally section 5 concludes the paper.

2. MOTIVATION

In this section, some of the existing sentiment classification approaches along with their benefits and drawbacks are discussed, which motivate the researchers to develop a new classification mechanism for classifying the tweets.

2.1 Literature Survey

Various existing sentiment classification techniques are reviewed in this section. Stieglitz and Xuan, (2013) developed a social media analytics framework in the view of political context. It effectively summarized number of political issues from political institutions. It served as the toolset for storing, analyzing, summarizing and monitoring the user generated information from social media. This framework was not suitable for the contexts, like marketing and business. Chen, *et al.*, (2018) introduced an attention-based Long short term memory (LSTM) classifier to perform sentiment analysis using twitter data. This model was very effective to extract the sentiment based embeddings. It offers effective guidance in the attention mechanism and better results of sentiments and semantics. It failed to consider the bi-embedding framework. Meduru, *et al.*, (2017) developed a sentiment intensity analyzer for analyzing the emotions of users with the sentiment analysis. This method minimized the gap and effectively made inactive participation among the users. Here, the twitter is used as platform for analyzing the user’s emotion. It is highly complex to classify the sarcastic comments due to the absence of tone in emoji. Singhal, *et al.*, (2015) introduced an unsupervised hybrid model based on the context and semantic aware rules for detecting the opinion of users and to predict the election result. It obtained better performance of system. However, this method was failed to detect the relationship between two different words.

Hasan, *et al.*, (2018) developed a hybrid approach using the machine learning classifier for analyzing the reviews in the twitter data. This method obtained better classification result by analyzing the tweets, but it failed to compute the patters from the political partyed using the twitter reviews. Kamyab, *et al.*, (2018) introduced a text mining approach for analyzing the twitter data. This approach accurately understands the happiness, opinions, and the discomforts of people. However, this model was failed to analyze the comments of other languages. Su, *et al.*, (2017) developed a supervised machine learning classifier to analyze the content in social media. It increased the efficiency, validity, and the reliability and tracked the expression from twitter data. It failed to classify the individual posts or tweets. Trupthi, *et al.*, (2018) modeled a possibilistic fuzzy c-means (PFCM) approach to perform the sentiment analysis using twitter data. It effectively identified the clustering heads from twitter content. It increased the accuracy of sentiment classification. However, it failed to increase the rate of classification using the multi-objective framework.
2.2 Challenges

Some of the challenges faced by the existing sentiment classification techniques are discussed as follows:

- To develop the robust sentiment classification model is major challenge, as it requires specialized knowledge and feature template to deal with the online reviews (Dimitrova, 2018).
- Opinion or sentiment spam detection is the major issue faced by the sentiment classification approach. Unlike other spam, it is very complex to find the opinion spam from trained dataset (Liu, 2012).
- To use both the tweet and the word level sentiment label at the learning stages creates challenging issue in the sentiment learning environment (Xiong, et al., 2018).
- It is very necessary to find the interaction among target and the context during sentiment learning, but it poses a great challenge (Zhang, et al., 2018).
- However, the text can be expressed using the syntactic and the semantic model; hence the highly generative scheme is used to represent the data. To find the dictionary and the feature resources associated with the sentiment data poses very complex task (Nio and Murakami, 2018).
- The contextual polarity is the major challenging task to perform the sentiment analysis process, as the polarity of the words changes with respect to various contexts.

3. PROPOSED PENGUIN RIDER OPTIMIZATION ALGORITHM BASED DEEP RECURRENT NEURAL NETWORK FOR SENTIMENT CLASSIFICATION

Twitter based social media analysis is used to recognize the opinions and sentiments that implicit in the social media data. However, it is very difficult to perform the sentiment classification using raw tweets, as it contains stop words, negative and positive emojis. In this research, an effective sentiment classification method named PeROA-based Deep RNN is proposed to perform the classification using political reviews. The proposed sentiment classification approach includes four different stages, namely pre-processing, feature extraction, feature selection, and sentiment classification. Initially, the political reviews are collected from the dataset such that the political reviews are taken as the input to perform sentiment classification process. The input political review data is subjected to the pre-processing module, where the stemming words and the stop words are removed. The pre-processed political review data is passed to the feature extraction stage, where the features, like elongated words, hashtags, multiple exclamation marks, number of sentences, multiple question marks, and absence/presence of greeting words are effectively extracted from the political review data. Accordingly, the essential features are optimally selected from the extracted features using mutual information. Finally, the sentiment classification process is performed using the Deep RNN classifier, which is trained by the proposed PeROA. However, the PeROA is developed by integrating the PeSOA (Gheraibia, et al., 2018) with ROA (Binu and Kariyappa, 2018). Figure 1 portrays the schematic diagram of the proposed PeROA-based Deep RNN.

3.1 Get the Input Twitter Data

The input data used to perform the sentiment classification is the political reviews of twitter data collected from the twitter dataset. Let us consider the database as with $nE$ number of political reviews $D$, which is expressed as,

$$E = \{D_1, D_2, \ldots, D_i, \ldots, D_n\} ; \quad 1 \leq i \leq n$$

(1)
where, $E$ is the database, $n$ specifies the total number of political reviews, $D$ signifies the political review data, and $D_i$ is the data located at the $i^{th}$ index. Here, the input data $D_i$ is selected from the database to perform sentiment classification process.

### 3.2 Pre-Processing the Political Reviews

The input political review data $D_i$ selected from the database is passed to the pre-processing module, where the stop word removal as well as the stemming process is carried out to enhance the quality of data. Pre-processing means cleaning and preparing the sentiment data for sentiment classification. The major benefit of using stop word removal and the stemming process is to reduce the noise and facilitate the feature extraction.

**Stop word removal:** Stop word is the word that does not have any important sentimental on specific domain. It is the process of removing all common words, such as “am”, “all”, “above”, “an”, “about”, “any”, and so on. It is used to reduce the dimensionality of data. While performing sentiment classification, the non-sentiment words present in the data must be removed in order to obtain better performance.

**Stemming:** It is the process of removing derived or inflected words to their root or stem word. Only, the word that has relevant words in same form in terms of meaning and structure is represented by the root. After pre-processing the input data, the features associated with the pre-processed data are required to be extracted in such a way that the pre-processed data is represented as $C_i$ with the dimension of $[U \times V]$.

### 3.3 Sentiment-Based Feature Extraction from Political Reviews

Once the input data is pre-processed, the result obtained from the pre-processing module $C_i$ is fed as the input to the feature extraction stage. Feature extraction have significant role in sentiment classification process.
classification. In this stage, the features associated with the sentiment data is effectively extracted such that the features include elongated words, hashtags, multiple exclamation marks, absence or presence of greeting words, number of sentences, and multiple question marks (Cheng, et al., 2011).

**Elongated words:** One character that is repeated for more than two times in the word is called elongated words and is represented as $f_1$. 

**Hashtags:** It is used to mark certain words in the tweets in order to specify the sentiment or topic and is represented as $f_2$. 

**Multiple exclamation marks:** It is defined as the number of contiguous sequences of exclamation marks in the sentiment data and is represented as $f_3$. 

**Absence/presence of greeting words:** It is the feature that contains the presence or absence of greeting words, like ‘kind regards’, and ‘best regards’ such that this features is represented as $f_4$. 

**Number of sentences:** It is the number of sentences that contains the sentiment word and is denoted as $f_5$. 

**Multiple question marks:** It is defined as the number of contiguous sequences of question marks in the sentiment data and is represented as $f_6$. Finally, the features that are extracted from the pre-processed sentiment data are represented as $f$ that includes $f = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ with the dimension of $[U \times V]$, respectively.

### 3.4 Feature Selection Using Mutual Information

The features $f$ that are extracted from the sentiment data is passed as input to the feature selection module. Here, the essential and the unique features are optimally selected using mutual information. The mutual information is calculated based on the probability values of extracted features such that the mutual information calculated using $f$ is represented as $G$. In addition to the extracted feature $f$, let us consider an additional sentiment-based feature $z$ such that the probability value of extracted feature $f$ and the sentiment-based feature is represented as $P(f)$ and $P(z)$. Accordingly, the mutual information used to select the features is expressed as,

$$G = \log_2 \frac{P(f, z)}{P(f)P(z)}$$

where, $G$ denotes the selected features, and $P(f, z)$ represents the joint probability of feature $f$ and $z$, respectively. The features that are selected from $f$ is represented as $G$ with the dimension of $[U \times V]$, which is passed as the input to the Deep RNN classifier to perform sentiment classification.

### 3.5 Sentiment Classification Using Proposed Penguin Rider Optimization Algorithm Based Deep RNN

This section describes the process of sentiment classification using Deep RNN classifier. However, the Deep RNN classifier is trained by the proposed PeROA, which is designed by integrating the PeSOA (Gheraibia, et al., 2018) with the ROA (Binu and Kariyappa, 2018). The algorithmic features from both the optimization enable the classifier to produce outperform result, which in turn boosts the performance of sentiment classification. PeSOA is the metaheuristic algorithm that is used to solve the optimization problem using the foraging behavior of penguins.
**Solution encoding:** It is the representation of solution that is to be determined using the optimization algorithm using the fitness measure.

**Fitness function:** It is the function used to find the optimal classification result based on the output of classifier. Accordingly, the fitness with the minimal error value is declared as the best solution for sentiment classification. The fitness function is computed as,

\[
F = \frac{1}{V} \sum_{\sigma=1}^{V} S^{(p,\sigma)} - \omega_{\sigma}
\]  

where, \( F \) specifies the fitness measure, \( S^{(p,\sigma)} \) indicates the output of classifier, and \( \omega_{\sigma} \) represents the estimated output.

### 3.5.1 Architecture of Deep RNN

This section elaborates the architecture of Deep RNN classifier, which is used to perform the sentiment classification of political reviews. The features \( G \) that extracted from the tweets are fed as the input to the Deep RNN structure. However, Deep RNN (Inoue, et al., 2018) classifier is the network architecture that consists of number of recurrent hidden layers associated in the network hierarchy. In the Deep RNN structure, the recurrent connection only exists in the hidden layer. The major benefit of using Deep RNN is that it effectively performs the classification process even under varying feature length with respect to the sequence of information. The previous state information is passed as the input to the current prediction and continues the iteration with the information associated in the hidden state. The features of the recurrent layer make this classifier to be very effective in performing sentiment classification. Therefore, Deep RNN is considered as the best classifier to perform sentiment classification of political reviews. Figure 2 portrays the architecture of Deep RNN classifier.
The structure of Deep RNN classifier is made by assuming the input vector of $p^{th}$ layer at $s^{th}$ time as $G^{(p,s)} = \{G_{1}^{(p,s)}, G_{2}^{(p,s)}, \ldots, G_{l}^{(p,s)}\}$ and the output vector of $p^{th}$ layer at $s^{th}$ time as $S^{(p,s)} = \{S_{1}^{(p,s)}, S_{2}^{(p,s)}, \ldots, S_{l}^{(p,s)}, \ldots, S_{q}^{(p,s)}\}$, respectively. Accordingly, the pair of each element of the input and the output vectors are called as unit. Here, $l$ specifies the arbitrary unit number of $p^{th}$ layer, and $q$ denotes the total number of units of $p^{th}$ layer. In addition to the above parameter, the arbitrary unit number of $(p-1)^{th}$ layer is indicated as $a$ and the total number of units of $(p-1)^{th}$ layer is represented as $I$, respectively. At this instance, the input propagation weight from $(p-1)^{th}$ layer to $p^{th}$ layer is specified as, $W^{(p)} \in B^{q \times l}$, and the recurrent weight of $p^{th}$ layer is denoted as $w^{(p)} \in B^{q \times q}$. Here, $B$ represents the set of weights. Moreover, the components of input vector is mathematically represented as,

$$G_{l}^{(p,s)} = \sum_{b=1}^{l} x_{lb}^{(p)} S_{b}^{(p-1,s)} + \sum_{j}^{q} h_{lj}^{(p)} S_{l}^{(p,s-1)}$$

(4)

where, $x_{lb}^{(p)}$ and $h_{lj}^{(p)}$ are the elements of $W^{(p)}$ and $w^{(p)}$. $l'$ specifies arbitrary unit number of $p^{th}$ layer. However, the elements of output vector of $p^{th}$ layer is expressed as,

$$S_{l}^{(p,s)} = \alpha^{(p)} \left(G_{l}^{(p,s)}\right)$$

(5)

where, $\alpha^{(p)}$ specifies the activation function. Moreover, the activation functions, namely sigmoid function as $\alpha(G) = \tanh(G)$, logistic sigmoid function as $\alpha(G) = \frac{1}{1 + e^{-G}}$, and the rectified linear unit function (ReLU) as $\alpha(G) = \max(G, \eta)$ are the frequently used activation function.

To make the classification process simple, let us assume $\eta^{th}$ weight as $x_{\eta}^{(p)}$ and $\eta^{th}$ unit as $S_{\eta}^{(p-1,s)}$ and hence, the bias is indicated as,

$$S_{l}^{(p,s)} = \alpha^{(p)} \left(W^{(p)} S^{(p-1,s)} + w^{(p)} \cdot S^{(p,s-1)}\right)$$

(6)

Here, $S^{(p,s)}$ specifies the classification output.

### 3.5.2 Proposed Penguin Rider Optimization Algorithm (PeROA)

The training of Deep RNN classifier is carried out by the proposed PeROA, which is designed by incorporating the PeSOA (Gheraibia, et al., 2018) with the ROA (Binu and Kariyappa, 2018) . It is required to train the classifier using the optimization in order to achieve the effectiveness of classification performance. The significance of PeROA is regarding the fast convergence to local optimum of PeSOA and the optimal convergence of ROA. The penguins work as the team and perform the dives simultaneously to feed on fish. Here, two forms of penguin groups are identified namely inter group and intra group, which are used to design the diversification and intensification strategy. However, the penguins migrate to other places to habitat food. Moreover, PeROA uses the
fictional computing model based on the thoughts and the ideas of rider groups. ROA contains four rider groups namely, overtaker, follower, attacker, and bypass rider, who travel to the destination to win the race. However, the four rider groups are discussed as follows:

**Bypass rider:** It is the first group of rider that bypasses the leading path and reaches the target.

**Follower:** It is the one that follows or depends on the path of leading rider to win the race.

**Attacker:** It is an aggressive player that takes the location of rider with maximum speed in order to move towards target.

**Overtaker:** This kind of rider follows its own position and move towards the target with respect to the position of leading rider.

The algorithmic steps involved in the proposed PeROA are discussed as follows:

a) **Rider parameter initialization:** Let us initialize the population of four rider groups as $L$ such that their position are specified in the groups randomly and is mathematically expressed as,

$$A_u = \{A_u(j,c)\}; \quad 1 \leq j \leq \beta; \quad 1 \leq c \leq \tau$$  \hspace{1cm} (7)

where, $A_u(j,c)$ signifies the position of $j^{th}$ rider at time $u$, $\tau$ symbolizes the number of dimension of coordinates, and $\beta$ represents the total number of riders. Accordingly, the steering angle of $j^{th}$ rider is represented as $X$, accelerator of $j^{th}$ rider is indicated as $\gamma$, $\rho$ specifies the brake of $j^{th}$ rider, and the gear of vehicle is represented as $k$, respectively.

b) **Find the success rate:** After initializing the parameters of rider groups, it is required to compute the success rate, which is computed based on the fitness measure and is specified in the Eq. (3).

c) **Find the leading rider:** Success rate plays the vital role to determine the leading rider. Accordingly, the rider with the maximum success rate is specified as the leading rider. Since the rider update its location based on the time, the leading rider in the rider groups is not fixed.

d) **Position update of riders:** The rider updates its location at each set for finding the leading rider. Based on the characteristics, like gear, brake, and accelerator, the rider updates their location and such that the update process of each rider is explains as follows:

i) **Update procedure of bypass rider:** The bypass rider does not follow the path of leading rider, as it bypasses the common path. However, the position update equation of such rider is mathematically modeled as,

$$A_{u+1}^R(j,c) = \mu \left[ A_u(y,c) \ast \chi(c) + A_u(\vartheta,c) \ast \left[1 - \chi(c)\right]\right]$$  \hspace{1cm} (8)

where, $A_{u+1}^R(j,c)$ symbolizes the position update process of bypass rider $R$, $\mu$ specifies the random number that lies in the interval of $[0,1]$, $y$ represents the random number that ranges between 1 to $\beta$, and $\vartheta$ specifies the random number that lies between 1 and $\beta$, and $\chi$ represents the random number that ranges between 0 and 1, respectively.
ii) **Update procedure of follower:** It follows the location of leading rider and updates its location in order to move the target more effectively. The update procedure of follower is based on the coordinate selector in such a way that the update equation is expressed as,

\[ A_{u+1}(j,r) = A^K(K,r) + \cos(X_{j,r}^u) * A^K(K,r) * H_j^u \]  

\[ A_{u+1}(j,r) = A^K(K,r) \left(1 + \cos(X_{j,r}^u) * H_j^u\right) \]  

The above equation specifies the position update process of follower in ROA. The swimming course update equation of penguins of PeSOA is mathematically expressed as,

\[ A_{u+1}(j,r) = A_u(j,r) + J_u(j,r) * rand(\cdot) * \left[A'_{\text{localbest}} - A_u(j,r)\right] \]  

\[ A_{u+1}(j,r) = A_u(j,r) + J_u(j,r) * rand(\cdot) * A'_{\text{localbest}} - J_u(j,r) * rand(\cdot) A_u(j,r) \]  

\[ A'_{\text{localbest}} = \frac{A_{u+1}(j,r) - A_u(j,r) + J_u(j,r) * rand(\cdot) A_u(j,r)}{J_u(j,r) * rand(\cdot)} \]  

By substituting the local best solution of PeSOA of Eq. (13) in Eq. (10), the resultant equation is expressed as,

\[ A_{u+1}(j,r) = \frac{A_{u+1}(j,r) - A_u(j,r) + J_u(j,r) * rand(\cdot) A_u(j,r)}{J_u(j,r) * rand(\cdot)} \left[1 + \cos(X_{j,r}^u) * H_j^u\right] \]  

\[ A_{u+1}(j,r) = \frac{A_{u+1}(j,r)}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) - \frac{A_u(j,r)[1 - J_u(j,r) * rand(\cdot)]}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) \]  

\[ A_{u+1}(j,r) - \frac{A_{u+1}(j,r)}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) = - \frac{A_u(j,r)[1 - J_u(j,r) * rand(\cdot)]}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) \]  

\[ A_{u+1}(j,r) \left[1 - \frac{\left(1 + \cos(X_{j,r}^u) H_j^u\right)}{J_u(j,r) * rand(\cdot)}\right] = - \frac{A_u(j,r)[1 - J_u(j,r) * rand(\cdot)]}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) \]  

\[ A_{u+1}(j,r) \left[\frac{J_u(j,r) * rand(\cdot)}{J_u(j,r) * rand(\cdot)} - 1 - \cos(X_{j,r}^u) H_j^u\right] = - \frac{A_u(j,r)[1 - J_u(j,r) * rand(\cdot)]}{J_u(j,r) * rand(\cdot)} \left(1 + \cos(X_{j,r}^u) H_j^u\right) \]
\[ A_{n+1}(j,r) = \frac{J_u(j,r) \ast \text{rand}(\_)}{J_u(j,r) \ast \text{rand}(\_)} \left(1 - \cos(X^u_{j,r})H^u_j\right) \]

\[ A_{n+1}(j,r) = \frac{A_u(j,r)[J_u(j,r) \ast \text{rand}(\_)] - 1}{J_u(j,r) \ast \text{rand}(\_)} \left(1 + \cos(X^u_{j,r})H^u_j\right) \]

(19)

\[ A_{n+1}(j,r) = \frac{A_u(j,r)[J_u(j,r) \ast \text{rand}(\_)] - 1}{J_u(j,r) \ast \text{rand}(\_)} \left(1 - \cos(X^u_{j,r})H^u_j\right) \]

(20)

Here, the term \( H^u_j \) is expressed as,

\[ H^u_j = \delta^u \ast \left(1 / Q_{off}\right) \]

(21)

\[ \delta^u_j = \frac{1}{3} \left[k^u_j \ast Y^i_j + Y^\max + \gamma^u_j + (1 - \rho^u_j) \ast Y^j\right] \]

(22)

where, \( rK \) indicates the coordinate selector, \( A^K \) denotes the location of leading rider, is the index of leading rider, \( X^u_{j,r} \) denotes the steering angle of \( j^{th} \) rider, and \( H^u_j \) represents the distance travelled by \( j^{th} \) rider. \( \delta^u_j \) specifies the velocity of \( j^{th} \) rider, \( Q_{off} \) denotes the off time, \( k^u_j \) indicates the gear of \( j^{th} \) rider vehicle, \( \gamma^u_j \) represents the accelerator of \( j^{th} \) rider, \( \rho^u_j \) denotes the brake of \( j^{th} \) rider, \( Y^\max \) represents the maximum speed of \( j^{th} \) rider, \( Y^j_{\text{max}} \) specifies the speed limit of gear, and \( J_u(j,r) \) denotes the oxygen reserve of \( j^{th} \) solution in \( u^{th} \) group.

iii) **Update procedure of overtaker:** The overtaker updates its position using the factors, namely success rate, coordinate selector, and direction indicator. Accordingly, the position update process of overtaker is represented as,

\[ A^O^V_{n+1}(j,r) = A_u(j,r) + [T(j) \ast A^K(K,r)] \]

(23)

where, \( A_u(j,r) \) denotes the location of \( j^{th} \) rider, and \( T \) represents the direction indicator of \( j^{th} \) rider.

iv) **Update process of attacker:** It attempts to consume the location of leader and follows the procedure of follower to update the position. The equation used by the attacker to update its location is expressed as,

\[ A^{AT}_{n+1}(j,c) = A^K(K,c) + \left[\cos(X^u_{j,c}) \ast A^K(K,c) \ast H^u_j\right] \]

(24)

where, \( A^K(K,c) \) represents position of leading rider, \( X^u_{j,c} \) represents the steering angle of \( j^{th} \) rider, and \( H^u_j \) represents the distance travelled by \( j^{th} \) rider.
e) **Evaluating feasibility:** Once the update process of rider is completed, the fitness for each rider is evaluated in such a way that the rider who is leading in the race is replaced with the new position of rider. However, the rider who have the maximal success rate is accepted as leading rider.

f) **Rider parameter update:** It is significant to update the parameters of rider in order to compute the best solution. In addition to the rider characteristics, the activity counter $\lambda$ is added in the rider update process.

g) **Termination:** The above steps are repeated until the best solution is obtained for sentiment classification. Algorithm 1 portrays the pseudo code of the proposed PeROA-based Deep RNN for sentiment classification.

By integrating the concept of fictional computing model with the foraging behavior of penguins, the performance of sentiment classification is increased. The parametric features from both the optimization framework enable to achieve better result through the political reviews of twitter data. With the diversification and the intensification strategy, the tweets are accurately classified as positive or negative tweets.

### 4. RESULTS AND DISCUSSION

In this section, the results and discussion made by the proposed PeROA-based Deep RNN is elaborated.

#### 4.1 Experimental Setup

The experimentation of the proposed approach is done in the PYTHON tool using the dataset employed in Twitter data: Pakistan election 2018 Twitter data: Pakistan elections 2018 taken from. (2012). https://www.kaggle.com/mohdazfar/pakistan-elections-2018. This dataset contains Twitter data of tweets that are related to the general election in Pakistan 2018. It consists of 29,579 data, and 46318 hashtags are available. Accordingly, the unique hashtags that are selected to perform the sentiment classification process is about 1946, respectively. However, the tweet contains number of attributes such that the text is selected based on the hashtags and count in order to classify the tweets as positive or negative. Here, the positive tweets are specified with the value of ‘1’, while the negative tweets are indicated with the value of ‘0’. 

#### 4.2 Evaluation Metrics

The performance of the sentiment classification approach is analyzed using the metrics, like sensitivity, accuracy, and specificity.

**Accuracy:** It is the ratio of number of correct detected assessments to the total number of assessments and is represented as,

$$\kappa = \frac{N_a + N_b}{N_a + N_b + O_a + O_b}$$  \hspace{1cm} (25)

where, $N_a$ specifies the true positive, $N_b$ is the true negative, $O_a$ denotes false positive, $O_b$ represents false negative, and $\kappa$ indicates accuracy.

**Sensitivity:** It is the ability of the measure for classifying the data to detect the true positive value and is expressed as,
Algorithm 1. Pseudo code of the proposed PeROA-based Deep RNN for sentiment classification

| Sl. No | Pseudo code of the proposed PeROA-based Deep RNN |
|--------|-----------------------------------------------|
| 1      | **Input**: Riders random position $A_n$        |
| 2      | **Output**: Leading rider $A^K$               |
| 3      | Begin                                         |
| 4      | Initialize riders population $L$              |
| 5      | Specify $\gamma$, $\rho$, $X$, and $k$        |
| 6      | Compute $F$                                   |
| 7      | while $u < Q_{off}$ : $Q_{off}$ -off time     |
| 8      | for $j = 1$ to $\beta$                      |
| 9      | Update $A^R_{n+1}(j,c)$ using Eq. (8)        |
| 10     | Update $A_{n+1}(j,r)$ using Eq. (20)         |
| 11     | Update $A^{OV}_{n+1}(j,r)$ using Eq. (23)    |
| 12     | Update $A^{AT}_{n+1}(j,c)$ using Eq. (24)    |
| 13     | Rank the riders based on $F$                 |
| 14     | Find the rider with maximum value of $F$ as leading rider. |
| 15     | Update $\gamma$, $\rho$, $X$, and $k$        |
| 16     | Return $A^K$                                 |
| 17     | $u = u + 1$                                  |
| 18     | end for                                      |
| 19     | end while                                   |
| 20     | terminate                                    |
\[ \mathcal{R} = \frac{N_a}{N_a + O_b} \]  

(26)

Here, \( \mathcal{R} \) denotes sensitivity.

**Specificity:** It is the ability of the measure for classifying the data to detect the true negative value and is expressed as,

\[ \nu = \frac{N_b}{N_b + O_a} \]  

(27)

where, \( \nu \) represents specificity.

**FPR:** It is the probability of rejecting the null values of tweets in sentiment classification and is represented as,

\[ \Upsilon = \frac{O_a}{O_a + N_b} \]  

(28)

Here, \( \Upsilon \) denotes FPR.

**FDR:** It is the measure that classifies the tweets as positive, but it holds the negative values and is represented as,

\[ Z = \frac{O_a}{O_a + N_a} \]  

(29)

Here, \( Z \) is the FDR.

**Precision:** It quantifies the number of positive predicted result that actually belongs to the positive class.

\[ P = \frac{N_a}{N_a + O_a} \]  

(30)

where, \( P \) denotes the precision.

**Recall:** It quantifies the number of positive prediction that is made out from the positive tweets.
\[ \Psi = \frac{N_a}{N_a + O_b} \]  

(31)

Here, \( \Psi \) denotes recall.

**F-measure**: It offers a single score that balances the concerns of recall and precision.

\[ \xi = \frac{2 \cdot \frac{P}{P + \Psi} \cdot \Psi}{P + \Psi} \]  

(32)

where, \( \xi \) denotes the F-measure.

### 4.3 Comparative Methods

The performance revealed by the PeROA-based Deep RNN approach is compared with the existing methods, like KNN (Huq, *et al*., 2017), social media analytics (Stieglitz and Xuan, 2013), sentiment intensity analyzer (Meduru, *et al*., 2017), and Attention-based Long Short Term Memory (LSTM) (Chen, *et al*., 2018), respectively.

### 4.4 Comparative Analysis

This section describes the analysis of the proposed PeROA-based Deep RNN in terms of the evaluation metrics by varying the training data. Figure 3 a) represents the comparative analysis of accuracy with that of training data. If training data=40%, the accuracy achieved by the corresponding KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.030%, 86.047%, 86.688%, and 87.328%, while the proposed PeROA-based Deep RNN obtained the accuracy of 89.1%, respectively. If training data=50%, the accuracy of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.407%, 86.311%, 87.441%, and 88.082%, while the proposed PeROA-based Deep RNN obtained the accuracy of 89.758%, respectively. If training data=70%, the accuracy of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 86.424%, 87.064%, 89.496%, and 90.052%, while the proposed PeROA-based Deep RNN obtained the accuracy of 91.591%, respectively.

Figure 3 b) portrays the analysis of sensitivity with that of training data. If training data=40%, the sensitivity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.185%, 85.639%, 85.903%, and 86.672%, while the proposed PeROA-based Deep RNN obtained the sensitivity of 89.100%, respectively. By considering 50% training data, the sensitivity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.207%, 85.750%, 86.873%, and 89.029%, while the proposed PeROA-based Deep RNN obtained the sensitivity of 89.743%, respectively. By considering 90% training data, the sensitivity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 87.599%, 90.131%, 90.684%, and 91.233%, while the proposed PeROA-based Deep RNN obtained the sensitivity of 92.030%, respectively.

Figure 3 c) depicts the analysis of specificity with that of training data. By considering 50% training data, the specificity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 80.185%, 85.639%, 85.903%, and 86.672%, while the proposed PeROA-based Deep RNN obtained better specificity of 90.100%, respectively. By considering 50% training data, the specificity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.207%, 85.750%, 86.873%, and 89.029%, while the proposed PeROA-based Deep RNN obtained the specificity of 89.743%, respectively. By considering 90% training data, the specificity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 87.599%, 90.131%, 90.684%, and 91.233%, while the proposed PeROA-based Deep RNN obtained the specificity of 92.030%, respectively.

Figure 3 d) depicts the analysis of specificity with that of training data. By considering 50% training data, the specificity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 80.185%, 85.639%, 85.903%, and 86.672%, while the proposed PeROA-based Deep RNN obtained better specificity of 90.100%, respectively. When By considering 70% training data, the specificity of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 83.223%, 89.496%, 90.169%, and 90.720%, while the proposed PeROA-based Deep RNN obtained better specificity of 91.734%, respectively. By considering 90% training data, the specificity of KNN, social media analytics, sentiment
intensity analyzer, and attention-based LSTM is 89.642%, 90.288%, 90.961%, and 92.030%, while the proposed PeROA-based Deep RNN obtained the specificity of 92.235%, respectively.

Figure 3 d) represents the analysis of recall with that of training data. By considering 60% training data, the recall of KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 85.326%, 85.757%, 88.704%, and 89.591%, while the proposed PeROA-based Deep RNN obtained better recall of 90.565%, respectively. By considering 80% training data, the recall obtained by the KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 86.784%, 89.972%, 90.288%, and 90.961%, while the proposed PeROA-based Deep RNN obtained better recall of 91.634%, respectively. If training data=90%, the recall obtained by the KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 87.599%, 90.131%, 90.684%, and 91.233%, while the proposed PeROA-based Deep RNN obtained better recall of 92.030%, respectively.

Figure 4 a) portrays the analysis of F-measure with that of training data. If training data=50%, the F-measure achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 86.958%, 87.459%, 88.704%, and 89.897%, while the proposed PeROA-based Deep RNN obtained better F-measure of 90.011%, respectively. When training data=70%, the F-measure achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 88.051%, 88.563%, 89.496%, and 90.108%, while the proposed PeROA-based Deep RNN obtained better F-measure of 91.634%, respectively. When training data=90%, the F-measure achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-
based LSTM is 90.045%, 90.684%, 91.264%, and 91.630%, while the proposed PeROA-based Deep RNN obtained better F-measure of 92.030%, respectively.

Figure 3(b). Comparative analysis b) sensitivity

![Comparative analysis of FPR and NPV](image)

Figure 4 c) depicts the analysis of NPV by varying training data. If training data=50%, the NPV achieved by the KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 863.397%, 66.343%, 77.154%, and 84.422%, while the proposed PeROA-based Deep RNN obtained better NPV of 89.100%, respectively. When training data=90%, the NPV achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 75.60%, 77.545%, 90.288%, and 90.961%, while the proposed PeROA-based Deep RNN obtained better NPV of 92.030%, respectively.

Figure 4 d) represents the analysis of FPR with that of training data. If training data=50%, the FPR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 18.837%, 13.505%, 10.405%, and 8.305%, while the proposed PeROA-based Deep RNN obtained lower FPR of 6.205%, respectively. If training data=80%, the FPR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 12.432%, 12.005%, 9.305%, and 6.705%, while the proposed PeROA-based Deep RNN obtained lower FPR of 4.605%, respectively. When training data=90%, the FPR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 11.505%, 9.405%, 7.305%, and 5.205%, while the proposed PeROA-based Deep RNN obtained lower FPR of 3.105%, respectively.
Figure 3(c). Comparative analysis c) specificity

Figure 3(d). Comparative analysis d) recall
Figure 4(a). Comparative analysis, a) F-measure

Figure 4(b). Comparative analysis, b) thread score
Figure 4(c). Comparative analysis, c) NPV

![NPV Comparison Graph](image)

Figure 4(d). Comparative analysis, d) FPR

![FPR Comparison Graph](image)
Figure 5(a) portrays the analysis of FNR with that of training data. If training data=50%, the FNR achieved by the KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 13.51%, 11.41%, 10.41%, and 8.35%, while the proposed PeROA-based Deep RNN obtained lower FNR of 5.58%, respectively. When training data=70%, the FNR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 12.51%, 12.51%, 10.41%, 8.81%, and 6.71%, while the proposed PeROA-based Deep RNN obtained lower FNR of 4.11%, respectively. When training data=90%, the FNR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 11.51%, 9.41%, 7.31%, and 5.63%, while the proposed PeROA-based Deep RNN obtained lower FNR of 3.11%, respectively.

Figure 5(b) depicts the analysis of FDR with that of training data. If training data=50%, the FDR achieved by the KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 13.505%, 11.405%, 10.405%, and 8.305%, while the proposed PeROA-based Deep RNN obtained lower FDR of 5.105%, respectively. When training data=70%, the FDR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 12.505%, 12.505%, 10.405%, 8.305%, and 6.205%, while the proposed PeROA-based Deep RNN obtained lower FDR of 4.105%, respectively. When training data=90%, the FDR achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 11.505%, 9.405%, 7.305%, and 5.205%, while the proposed PeROA-based Deep RNN obtained lower FDR of 3.105%, respectively.
4.5 Comparative Discussion

Table 1 portrays the comparative discussion of the proposed model. For 90%, the sensitivity obtained by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 87.599%, 90.131%, 90.684%, and 91.233%, while the proposed PeROA-based Deep RNN obtained better sensitivity of 92.030%, respectively. For 90% training data, the specificity achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 89.642%, 90.288%, 90.961%, and 92.030%, while the proposed PeROA-based Deep RNN obtained better specificity of 92.235%, respectively. If training data=90%, the F-measure achieved by the existing, KNN, social media analytics, sentiment intensity analyzer, and attention-based LSTM is 90.045%, 90.684%, 91.264%, and 91.630%, while the proposed PeROA-based Deep RNN obtained better F-measure of 92.030%, respectively. It is very clear that the proposed PeROA-based Deep RNN obtained better performance in sentiment classification.
5. CONCLUSION

In this research, an effective sentiment classification approach named PeROA-based Deep RNN is proposed for classifying the sentiment reviews as either positive or negative tweet. However, the sentiment classification strategy is performed by considering the political reviews of twitter data as input. At first, the input political review data is fed to the pre-processing module, where the stop word removal and the stemming process are progressed to reduce the noise of data. In the pre-processing stage, the dimensionality of data gets reduced. The pre-processed data is subjected to the feature extraction module, where the sentiment based features are effectively extracted from the pre-processed result. Accordingly, the unique features are effectively selected from the data using mutual information. Finally, the sentiment classification process is achieved using the Deep RNN classifier, which is trained by the proposed PeROA. However, the PeROA is developed by integrating the PeSOA with the ROA. Moreover, the proposed PeROA-based Deep RNN obtained better performance with the metrics, like accuracy, sensitivity, specificity, recall, F-measure, thread score, NPV, FPR, FNR, and FDR with the values of 92.030%, 92.030%, 92.235%, 92.030%, 92.030%, 92.030%, 3.105%, 3.11%, and 3.105%, respectively. In future, the performance of sentiment classification can be further increased using some other optimization framework.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.
REFERENCES

Bennett, W. L., & Pfetsch, B. (2018). Rethinking Political Communication in a Time of Disrupted Public Spheres. *Journal of Communication, 68*(2), 243–253.

Binu, D., & Kariyappa, B. S. (2018). RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits. *IEEE Transactions on Instrumentation and Measurement*.

Bode, L. (2016). Political News in the News Feed: Learning Politics From Social Media. *Mass Communication & Society, 19*(1), 24–48.

Chen, Y., Yuan, J., You, Q., & Luo, J. (2018). *Twitter Sentiment Analysis via Bi-sense Emoji Embedding and Attention-based LSTM*. Academic Press.

Cheng, N., Chandramouli, R., & Subbalakshmi, K. P. (2011). Author gender identification from text. *Digital Investigation, 8*, 78–88.

Dimitrova, D. V. (2018). *Social Media in Political Campaigning Around the World: Theoretical and Methodological Challenges*. Academic Press.

Dingli, A., Mercieca, L., Spina, R., & Galea, M. (2015). Event detection using social sensors. *Proceedings of 2nd International Conference on Information and Communication Technologies for Disaster Management*.

Fu, X., Wei, Y., Xu, F., Wang, T., Lu, Y., Li, J., & Huang, J. Z. (2019). Semi-supervised aspect-level sentiment classification model based on variational autoencoder. *Knowledge-Based Systems, 171*, 81–92.

Gheraibia, Y., Moussaoui, A., Yin, P. Y., Papadopoulos, Y., & Maazouzi, S. (2018). *PeSOA: Penguins Search Optimisation Algorithm for Global Optimisation Problems*. arXiv preprint arXiv:1809.09895.

Glowacki, M., Narayanan, V., Maynard, S., & Hirsch, G. (2018). *News and Political Information Consumption in Mexico: Mapping the 2018 Mexican Presidential Election on Twitter and Facebook*. Academic Press.

Guerrero, J. S., Olivas, J. A., Romero, F. P., & Viedma, E. H. (2015). Sentiment analysis: A review and comparative analysis of web services. *Information Sciences, 311*, 18–38.

Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine Learning-Based Sentiment Analysis for Twitter Accounts. *Mathematical and Computational Applications, 23*(1), 11.

Huang, B., & Carley, K. M. (2019). Parameterized convolutional neural networks for aspect level sentiment classification. arXiv preprint arXiv:1909.06276.

Huq, M. R., Ali, A., & Rahman, A. (2017). Sentiment analysis on Twitter data using KNN and SVM. *International Journal of Advanced Computer Science and Applications, 8*(6), 19–25.

Inoue, M., Inoue, S., & Nishida, T. (2018). Deep recurrent neural network for mobile human activity recognition with high throughput. *Artificial Life and Robotics, 23*(2), 173–185.

Joshi, O. S., & Simon, G. (2018). Sentiment Analysis Tool on Cloud: Software as a Service Model. *Proceedings of International Conference On Advances in Communication and Computing Technology (ICACCT)*.

Kanyab, M., Tao, R., Mohammadi, M. H., & Rasool, A. (2018). *Sentiment Analysis on Twitter: A text Mining Approach to the Afghanistan Status Reviews*. Academic Press.

Kanavos, A., Perikos, I., Hatzilygeroudis, I., & Tsakalidis, A. (2017). Emotional community detection in social networks. *Computers & Electrical Engineering, 1–12.*

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Academic Press.

Ma, R., Wang, K., Qiu, T., Sangaiah, A. K., Lin, D., & Liaqat, H. B. (2019). Feature-based compositing memory networks for aspect-based sentiment classification in social internet of things. *Future Generation Computer Systems, 92*, 879–888.

Mattila, M., & Salman, H. (2018). *Analysing Social Media Marketing on Twitter using Sentiment Analysis*. Academic Press.
Meduru, M., Mahimkar, A., Subramanian, K., Padiya, P. Y., & Gunjgur, P. N. (2017). Opinion Mining Using Twitter Feeds for Political Analysis. *International Journal of Computer Science and Engineering*, 25(1), 116–123.

Nio, L., & Murakami, K. (2018). Japanese Sentiment Classification Using Bidirectional Long Short-Term Memory Recurrent. *Neural Networks*, 1119–1122.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. *Proceedings of 19th International Conference on World Wide Web*.

Si, S. (2016). Social Media and Its Role in Marketing. *Business and Economics Journal*, 7(1).

Singhal, K., Agrawal, B., & Mittal, N. (2015). Modeling Indian General Elections: Sentiment Analysis of Political Twitter Data. *Information Systems Design and Intelligent Applications*, 469–477.

Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019). Attentional encoder network for targeted sentiment classification. arXiv preprint arXiv:1902.09314.

Stieglitz, S., & Xuan, L.D. (2013). *Social media and political communication: A social media analytics framework*. Academic Press.

Stromback, J. (2017). Does public service TV and the intensity of the political information environment matter? *Journalism Studies*, 18(11), 1415–1432.

Su, L. Y. F., Cacciatoro, M. A., Liang, X., Brossard, D., Scheufele, D. A., & Xenos, M. A. (2017). Analyzing public sentiments online: Combining human- and computer-based content analysis. *Information Communication and Society*, 20(3), 406–427.

Trupthi, M., Pabboju, S., & Narsimha, G. (2018). Possibilistic Fuzzy C-means Topic Modelling for Twitter Sentiment Analysis. *International Journal of Intelligent Engineering and Systems*, 11(3), 100–108.

Twitter data Pakistan elections 2018. (2022). https://www.kaggle.com/mohdazfar/pakistan-elections-2018

Xiong, S., Lv, H., Zhao, W., & Ji, D. (2018). Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings. *Neurocomputing*, 275, 2459–2466.

Xue, Q., Zhang, W., & Zha, H. (2020). *Improving Domain-Adapted Sentiment Classification by Deep Adversarial Mutual Learning*. arXiv preprint arXiv:2002.00119.

Yang, M., Yin, W., Qu, Q., Tu, W., Shen, Y., & Chen, X. (2019). Neural attentive network for cross-domain aspect-level sentiment classification. *IEEE Transactions on Affective Computing*.

Zhang, Y., Miao, D., Wang, J., & Zhang, Z. (2019). A cost-sensitive three-way combination technique for ensemble learning in sentiment classification. *International Journal of Approximate Reasoning*, 105, 85–97.

Zhang, Y., Zhang, Z., Miao, D., & Wang, J. (2019). Three-way enhanced convolutional neural networks for sentence-level sentiment classification. *Information Sciences*, 477, 55–64.

Zhang, Z., Wang, L., Zou, Y., & Gan, C. (2018). The optimally designed dynamic memory networks for targeted sentiment classification Zufan. *Neurocomputing*.