RESEARCH

Open Access

Twitter sentiment analysis using hybrid gated attention recurrent network

Nikhat Parveen[1,2*], Prasun Chakrabarti[3], Bui Thanh Hung[4] and Amjan Shaik[2,5]

*Correspondence:
nikhat0891@gmail.com

[1] Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Guntur-Dt, Vaddeswaram, Andhra Pradesh, India
[2] Ton Duc Thang University, Ho Chi Minh, Vietnam
[3] ITM SLS Baroda University, Vadodara, Gujarat, India
[4] Data Science Laboratory, Faculty of Information Technology, Industrial University of Ho Chi Minh, Ho Chi Minh, Vietnam
[5] Department of Computer Science & Engineering, St.Peter's Engineering College, Hyderabad, India

Abstract

Sentiment analysis is the most trending and ongoing research in the field of data mining. Nowadays, several social media platforms are developed, among that twitter is a significant tool for sharing and acquiring peoples' opinions, emotions, views, and attitudes towards particular entities. This made sentiment analysis a fascinating process in the natural language processing (NLP) domain. Different techniques are developed for sentiment analysis, whereas there still exists a space for further enhancement in accuracy and system efficacy. An efficient and effective optimization based feature selection and deep learning based sentiment analysis is developed in the proposed architecture to fulfil it. In this work, the sentiment 140 dataset is used for analysing the performance of proposed gated attention recurrent network (GARN) architecture. Initially, the available dataset is pre-processed to clean and filter out the dataset. Then, a term weight-based feature extraction termed Log Term Frequency-based Modified Inverse Class Frequency (LTF-MICF) model is used to extract the sentiment-based features from the pre-processed data. In the third phase, a hybrid mutation-based white shark optimizer (HMWSO) is introduced for feature selection. Using the selected features, the sentiment classes, such as positive, negative, and neutral, are classified using the GARN architecture, which combines recurrent neural networks (RNN) and attention mechanisms. Finally, the performance analysis between the proposed and existing classifiers is performed. The evaluated performance metrics and the gained value for such metrics using the proposed GARN are accuracy 97.86%, precision 96.65%, recall 96.76% and f-measure 96.70%, respectively.

Keywords: Deep learning, Term weight-feature extraction, White shark optimizer, Twitter sentiment, Gated recurrent attention network, Natural language processing, Recurrent neural network

Introduction

Sentiment Analysis (SA) uses text analysis, NLP (Natural Language Processing), and statistics to evaluate the user's sentiments. SA is also called emotion AI or opinion mining [1]. The term 'sentiment' refers to feelings, thoughts, or attitudesexpressed about a person, situation, or thing. SA is one of the NLP techniques used to identify whether the obtained data or information is positive, neutral or negative. Business experts frequently use it to monitor or detect sentiments to gauge brand reputation, social data and understand customer needs [2, 3]. Over recent years, the amount of information uploaded or

© The Author(s) 2023, corrected publication 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

generated online has rapidly increased due to the enormous number of Internet users [4, 5].

Globally, with the emergence of technology, social media sites [6, 7] such as Twitter, Instagram, Facebook, LinkedIn, YouTube etc.,have been used by people to express their views or opinions about products, events or targets. Nowadays, Twitter is the global micro-blogging platform greatly preferred by users to share their opinions in the form of short messages called tweets [8]. Twitterholds 152 M (million) daily active users and 330 M monthly active users,with 500 M tweets sent daily [9]. Tweets often effectively createa vast quantity of sentiment data based on analysis. Twitter is an effective OSN (online social network) for disseminating information and user interactions. Twitter sentiments significantly influence diverse aspects of our lives [10]. SA and text classification aims at textual information extraction and further categorizes the polarity as positive (P), negative (N) or neutral (Ne).

NLP techniques are often used to retrieve information from text or tweet content. NLP-based sentiment classification is the procedure in which the machine (computer) extracts the meaning of each sentence generated by a human. Manual analysis of TSA (Twitter Sentiment Analysis) is time-consuming and requires more experts for tweet labelling. Hence, to overcome these challenges automated model is developed. The innovations of ML (Machine learning) algorithms [11, 12],such as SVM (Support Vector Machine), MNB (Multinomial Naïve Bayes), LR (Logistic Regression), NB (Naïve Bayes) etc., have been used in the analysis of online sentiments. However, these methods illustrated good performance, but these approaches are very slow and need more time to perform the training process.

DL model is introduced to classify Twitter sentiments effectively. DL is the subset of ML that utilizes multiple algorithms to solve complicated problems. DL uses a chain of progressive events and permits the machine to deal with vast data and little human interaction. DL-based sentiment analysis offers accurate results and can be applied to various applications such as movie recommendations, product predictions, emotion recognition [13–15],etc. Such innovations have motivated several researchers to introduce DL in Twitter sentiment analysis.

Motivation

SA (Sentiment Analysis) is deliberated with recognizing and classifying the polarity or opinions of the text data. Nowadays, people widely share their opinions and sentiments on social sites. Thus, a massive amount of data is generated online, and effectively mining the online data is essential for retrieving quality information. Analyzing online sentiments can createa combined opinion on certain products. Moreover, TSA (Twitter Sentiment Analysis) is challenging for multiple reasons. Short texts (tweets), owing to the maximum character limit, is a major issue. The presence of misspellings, slang and emoticons in the tweets requires an additional pre-processing step for filtering the raw data. Also, selecting a new feature extraction model would be challenging,further impacting sentiment classification. Therefore, this work aims to develop a new feature extraction and selection approach integrated with a hybrid DL classification model for accurate tweet sentiment classification. The existing research works [16–21] focus on DL-based TSA, which haven't attained significant results because of smaller dataset usage and slower manual text labelling. However, the datasets with unwanted details and spaces also reduce the classification algorithm's efficiency. Further, the dimension occupied by extracted features also degrades the efficiency of a DL approach. Hence, to overcome such issues, this work aims to develop a successful DL algorithm for performing Twitter SA. Pre-processing is a major contributor to this architecture as it can enhance DL efficiency by removing unwanted details from the dataset. This pre-processing also reduces the processing time of a feature extraction algorithm. Followed to that, an optimization-based feature selection process was introduced, which reduces the effort of analyzing irrelevant features. However, unlike existing algorithms, the proposed GARN can efficiently analyse the text-based features. Further, combining the attention mechanism with DL has enhanced the overall efficiency of the proposed DL algorithm. As attention mechanism have the greater ability to learn the selected features by reducing the complexity of model. This merit causes the attention mechanism to integrate with RNN and achieved effective performance.

Objectives

The major objectives of the proposed research are:

- To introduce a new deep model Hybrid Mutation-based White Shark Optimizer with a Gated Attention Recurrent Network (HMWSO-GARN) for Twitter sentiment analysis.
- The feature set can be extracted with the new Term weighting-based feature extraction (TW-FE) approach named Log Term Frequency-based Modified Inverse Class Frequency (LTF-MICF) is used and compared with traditional feature extraction models.
- To identify the polarity of tweets with the bio-inspired feature selection and deep classification model.
- To evaluate the performance using different metrics and compare it with traditional DL procedures on TSA.

Related works

Some of the works related to DL-based Twitter sentiment analysis are:

Alharbi et al. [16] presented the analysis of Twitter sentiments using a DNN (deep neural network) based approach called CNN (Convolutional Neural Network). The classification of tweets was processed based on dual aspects, such as using social activities and personality traits. The sentiment (P, N or Ne) analysis was demonstrated with the CNN model, where the input layer involves the feature lists and the pre-trained word embedding (Word2Vec). The dual datasets used for processing were SemEval-2016_1 and SemEval-2016_2. The accuracy obtained by CNN was 88.46%, whereas the existing methods achieved less accuracy than CNN. The accuracy of existing methods is LSTM (86.48%), SVM (86.75%), KNN (k-nearest neighbour) (82.83%), and J48 (85.44%), respectively.

Tam et al. [17] developed a Convolutional Bi-LSTM model based on sentiment classification on Twitter data. Here, the integration of CNN-Bi-LSTM was characterized byextracting local high-level features. The input layer gets the text input and slices it into tokens. Each token was transformed into NV (numeric values). Next, the pre-trained WE (word embedding), such as GloVe and W2V (word2vector), were used to create the word vector matrix. The important words were extracted using the CNN model,and the feature set was further minimized using the max-pooling layer. The Bi-LSTM (backwards, forward) layers were utilized to learn the textual context. The dense layer (DeL) was included after the Bi-LSTM layer to interconnect the input data with output using weights. The performance was experimented using datasets TLSA (Twitter Label SA) and SST-2 (Stanford Sentiment Treebank). The accuracy with the TLSA dataset was (94.13%) and (91.13%) with the SST-2 dataset.

Chugh et al. [18] developed an improved DL model for information retrieval and classification of sentiments. The hybridized optimization algorithm SMCA was the integration of SMO (Spider Monkey Optimization) and CSA (Crow Search Algorithm). The presented DRNN (DeepRNN) was trained using the algorithm named SMCA. Here, the sentiment categorization was processed with DeepRNN-SMCA and the information retrieval was done with FuzzyKNN. The datasets used were the mobile reviews amazon dataset and telecom tweets dataset. Forsentiment classification, the accuracy obtained on the first dataset was (0.967), andthe latter was gained (0.943). The performance with IR (information retrieval) on dataset 1 gained (0.831) accuracy and dataset 2 obtained (0.883) accuracy.

Alamoudi et al. [19] performed aspect-based SA and sentiment classification aboutWE (word embeddings) and DL. The sentiment categorization involves both ternary and binary classes. Initially, the YELP review dataset was prepared and pre-processed for classification. The feature extraction was modelled with TF-IDF, BoW and Glove WE. Initially, the NB and LR were used for first set feature (TF-IDF, BoW features) modelling; then, the Glove features were modelled using diverse models such as ALBERT, CNN, and BERT for the ternary classification. Next, aspect and sentence-based binary SA was executed. The WE vector for sentence and aspect was done with the Glove approach. The similarity among aspects and sentence vectors was measured using cosine similarity, and binary aspects were classified. The highest accuracy (98.308%) was obtained when executed with the ALBERT model on aYELP 2-class dataset, whereas the BERT model gained (89.626%) accuracy with a YELP 3-class dataset.

Tan et al. [20] introduced a hybrid robustly optimized BERT approach (RoBERTa) with LSTM for analyzing the sentiment data with transformer and RNN. The textual data was processed with word embedding, and tokenization of the subwordwas characterized with the RoBERTa model. The long-distance Tm (temporal) dependencies were encoded using the LSTM model. The DA (data augmentation) based on pre-trained word embedding was developed to synthesize multiple lexical samples and present the minority class-based oversampling. Processing of DA solves the problem of an imbalanced dataset with greater lexical training samples. The Adam optimization algorithm was used to perform hyperparameter tuning,leading to greater results with SA. The implementation datasets were Sentiment140,Twitter US Airline,and IMDb datasets. The overall accuracy gained with these datasets was 89.70%, 91.37% and 92.96%, respectively.

Hasib et al. [21] proposed a novel DL-based sentiment analysis of Twitter data for the US airline service. The Twitter tweet is collected from the Kaggle dataset: crowdflowerTwitter US airline sentiment. Two models are used for feature extraction:DNN and convolutional neural network (CNN). Before applying four layers, the tweets are converted to metadata and tf-idf. The four layers of DNN aretheinput, covering, and output layers. CNN for feature extraction is by the following phases; data pre-processing, embedded features, CNN and integration features. The overall precision is 85.66%, recall is 87.33%, and f1-score is 87.66%, respectively. Sentiment analysis was used to identify the attitude expressed using text samples. To identify such attitudes, a novel term weighting scheme was developed by Carvalho and Guedes in [24], which was an unsupervised weighting scheme (UWS). It can process the input without considering the weighting factor. The SWS (Supervised Weighting Schemes) was also introduced, which utilizes the class information related to the calculated term weights. It had shown a more promising outcome than existing weighting schemes.

Learning from online courses are considered as the mainstream of learning domain. However, it was identified that analysing the users comments are considered as the major key for enhancing the efficiency and quality of online courses. Therefore, identifying sentiments from the user's comments were considered as the efficient process for enhancing the learning process of online course. By taking this as major goal, an ensemble learning architecture was introduced by Pu et al. in [34] which utilizes glove, and Word2Vec for obtaining vector representation. Then, the extraction of deep features was achieved using CNN (Convolutional neural network) and bidirectional long and short time network (Bi-LSTM). The integration of suggested models were achieved using ensemble multi-objective gray wolf optimization (MOGWO). It achieves 91% f1-score value.

The sentiment dictionaries use binary sentiment analysis like BERT, word2vec and TF-IDF were used to convert movie and product review into vectors. Three-way decision in binary sentiment analysis separates the data sample into uncertain region (UNC), positive (POS) region and Negative (NEG) region. UNC got benefit from this three-way decision model and enhances the effect of binary sentiment analysis process. For the optimal feature selection, Chen, J et al. [35] developed a three-way decision model which get the optimal features representation of positive and negative domains for sentiment analysis. Simulation was done in both Amazon and IMDB database to show the effectiveness of the proposed methodology.

The advancements in biga data analytics (BDA) model is obtained by the people who generate large amount of data in their day-to-day live. The linguistic based tweets, feature extraction and sentimental texts placed between the tweets are analysed by the sentimental analysis (SA) process. In this article, Jain, D.K et al. [36] developed a model which contains pre-processing, feature extraction, feature selection and classification process. Hadoop Map Reduce tool is used to manage the big data, then pre-processing method is initiated to remove the unwanted words from the text. For

Table 1 Twitter sentiment analysis using DL techniques

Author & year	Methodology	Merits	Demerits
Alharbi et al. 2019 [16]	CNN (Convolutional Neural Network)	The behavioural information of the user is included	Difficult to interpret the exact tweet from a group of tweets
Tam et al. 2021 [17]	Hybrid CNN-BiLSTM (Convolution neural network and bidirectional long short-term memory)	The performance of the word embedding techniques is high	Lower classification and retrieval accuracy
Chugh et al. 2021 [18]	DRNN (DeepRNN), SMO (Spider Monkey Optimization) and CSA (Crow Search Algorithm)	Provides better reviews to take effective decisions	Lower performance accuracy
Alamoudi et al. 2021 [19]	Convolutional neural network (CNN), BERT and ALBERT models	Reduction in error rate	Occurrence of mislabelled reviews
Tan et al. 2022 [20]	BERT approach (RoBERTa) with LSTM	Optimization is done using the word embedding technique	Lower classification accuracy
Hasib et al. 2021 [21]	CNN (Convolutional neural network) and DNN	Collected data on the emotions of the airline consumers	Less number of tweets are used
Guedes, G.P. 2020 [24]	UWS, SWS	Efficiency of proposed over the used dataset is found high	High error obtained

feature extraction, TF-IDF vector is utilized and Binary Brain Storm Optimization (BBSO) is used to select the relevant features from the group of vectors. Finally, the incidence of both positive and negative sentiments is classified using Fuzzy Cognitive Maps (FCMs). Table 1 shows the comparative analysis of Twitter sentiment analysis using DL techniques.

Problem statement

There are many problems related to twitter sentiment analysis using DL techniques. The author in [16] has used the DL model and performed the sentiment classification from Twitter data. To classify such data, this method analysed each user's behavioural information. However, this method has faced struggles in interpreting exact tweet words from the massive tweet corpus; due to this, the efficiency of a classification algorithm has been reduced.ConvBiLSTM was introduced in [17], which used glove and word2vec-based features for sentiment classification. However, the extracted features are not sufficient to achieve satisfactory accuracy. Then, processing time reduction was considered a major objective in [18], which utilizes DeepRNN for sentiment classification. But it fails to reduce the dimension occupied by the extracted features. This makes several valuable featuresfall within the local optimum. DL and word embedding processes were combined in [19], which utilizes Yelp reviews for processing. It has shown efficient performance for two classes but fails to provide better accuracy for three-class classification. Recently, a hybrid LSTM architecture was developed in [20], which has shown flexible processing over sentiment classification and takes a huge amount of time to process large datasets. DNN-based feature extraction and CNN-based sentiment classification were performed in [21], which haven't shown more efficient performance than other algorithms. Further, it also concentrated only on 2 classes.

Few of the existing literatures fails to achieve efficient processing time, complexity and accuracy due to the availability of large dataset. Further, the extraction of low-level and unwanted features reduces the efficiency of classifier. Further, the usage of all extracted features occupies large dimension. These demerits makes the existing algorithms not suitable for efficient processing. This shortcomings open a research space for efficient combined algorithm for twitter data analysis. To overcome such issue, the proposed architecture has combined RNN and attention mechanism. The features required for classification is extracted using LTF-MICF which provides features for twitter processing. Then, the dimension occupied by huge extracted features are reduced using HMWSO algorithm. This algorithm has the ability to process the features in less time complexity and shows better optimal feature selection process. This selected features has enhanced the performance of proposed classifier over the large dataset and also achieved efficient accuracy with less misclassification error rate.

Proposed methodology

For sentiment classification of Twitter tweets, a DL technique of gated attention recurrent network (GARN) is proposed. The Twitter dataset (Sentiment140 dataset) with sentiment tweets that the public can access is initially collected and given as input. After collecting data, the next stage is pre-processing the tweets. In the pre-processing stage, tokenization, stopwords removal, stemming, slang and acronym correction, removal of numbers, punctuations &symbol removal, removal of uppercase and replacing with lowercase, character &URL, hashtag & user mention removal are done. Now the pre-processed dataset act as input for the next process. Based on term frequency, a term weight is allocated for each term in the dataset using the Log Term Frequency-based Modified Inverse Class Frequency (LTF-MICF) extraction technique. Next, Hybrid Mutation based White Shark Optimizer (HMWSO) is used to select optimal term weight. Finally, the output of HMWSO is fed into the gated attention recurrent network (GARN) for sentiment classification with three different classes. Figure 1 shows a diagrammatic representation of the proposed methodology.

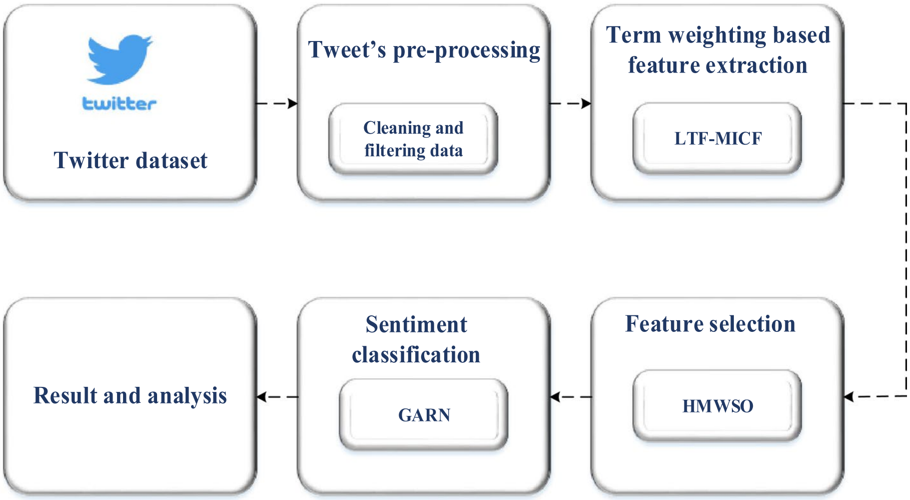


Fig. 1 Architecture diagram

Tweets pre-processing

Pre-processing is converting the long data into short text to perform other processes such as classification, detecting unwanted news, sentiment analysis etc., as Twitter users use different styles to post their tweets. Some may post the tweet in abbreviations, symbols, URLs, hashtags, and punctuations. Also, tweets may consist of emojis, emoticons, or stickers to express the user's sentiments and feelings. Sometimes the tweets may be in a hybrid form,such as adding abbreviations, symbols and URLs. So these kinds of symbols, abbreviations, and punctuations should be removed from the tweet toclassify the dataset further. The features to be removed from the tweet dataset are tokenization, stopwords removal, stemming, slag and acronym correction, removal of numbers, punctuation and symbol removal, noise removal, URL, hashtags, replacing long characters, upper case to lower case, and lemmatization.

Tokenization

Tokenization [28] is splitting a text cluster into small words, symbols, phrases and other meaningful forms known as tokens. These tokens are considered as input for further processing. Another important use of tokenization is that it can identify meaningful words. The tokenization challenge depends only on the type of language used. For example, in languages such as English and French, some words may be separated by white spaces. Other languages, such as Chinese and Thai words,are not separated. The tokenization process is carried out in the NLTK Python library. In this phase, the data is processed in three forms: convert the text document into word counts. Secondly,data cleansing and filtering occur, andfinally, the document is split into tokens or words.

The example provided below illustrates the original tweet before and after performing tokenization:

Before tokenization

DLis a technology which trains the machineto behave naturally like a human being.

After tokenization

Deep, learning, is, a, technology, which, train, the, machine, to, behave, naturally, like, a, human, being.

Numerous tools are available to tokenize a text document. Some of them are as follows;

- NLTK word tokenize
- Nlpdotnet tokenizer
- TextBlob word tokenize
- Mila tokenizer
- Pattern word tokenize
- MBSP word tokenize

Stopwords removal

Stopword removal [28] is a process of removing frequently used words with meaningless in a text document. Stopwords such as are, this, that, and, so are frequently occurring words in a sentence. These words are also termed pronouns, articles and prepositions. Such words are not used forfurther processing, so removing those words is required. If such words are not removed, the sentence seems heavy and becomes less important for the analyst.Also, they are not considered keywords in Twitter analysis applications. Many methods exist to remove stopwords from a document; they are.

- Z-methods
- Classic method
- Mutual information (MI) method
- Term based random sampling (TBRS) method

Removing stopwords from a pre-compiled list is performed using a classic-based method. Z-methods are known as Zipf's law-based methods. In Z-methods, three removal processes occur: removing the most frequently used words, removing the words which occur once in a sentence, and removing words with a document frequency of low inverse. In the mutual MI method, the information with low mutual will be removed. In the TBRS method, the words are randomly chosen from the document and given rank for a particular term using the Kullback–Leibler divergence formula, which is represented as;

$$d_l(t) = Q_l(t).\log_2 \frac{Q_l(t)}{Q(t)} \tag{1}$$

where $Q_l(t)$ is the normalized term frequency (NTF) of the term t within a mass l, and NTF is denoted as $Q(t)$ of term t in the entire document. Finally, using this equation, the least terms are considered a stopword list from which the duplications are removed.

Stemming

Removing prefixesand suffixes from a word is performed using the stemming method. It can also be defined as detecting the root and stem of a word and removing them. For example, processed word processing can be stemmed from a single word as a process [28]. The two points to be considered while performing stemming are: the words with different meanings must be kept separate, and the words of morphological forms will contain the same meaning and must be mapped with a similar stem. There are stemming algorithms to classify the words. The algorithms are divided into three methods: truncating, statistical, and mixed methods. Truncating method is the process of removing a suffix from a plural word. Some rules must be carried out to remove suffixes from the plurals to convert the plural word into the singular form.

Different stemmer algorithms are used under the truncating method. Some algorithms are Lovins stemmer, porters stemmer, paice and husk stemmer, and Dawson stemmer.

Lovins stemmer algorithm is used to remove the lengthy suffix from a word. The drawback of using this stemmer is that it consumes more time to process. Porter's stemmer algorithm removes suffixes from a word by applying many rules. If the applied rule is satisfied, the suffix is automatically removed. The algorithm consists of 60 rules and is faster than theLovins algorithm. Paice and husk is an iterative algorithm that consists of 120 rules to remove the last character of the suffixed word. This algorithm performs two operations, namely, deletion and replacement. The Dawson algorithm keeps the suffixed words in reverse order by predicting their length and the last character. In statistical methods, some algorithms are used: N-gram stemmer, HMM stemmer, and YASS stemmer. In a mixed process, the inflectional and derivational methods are used.

Slang and acronym correction

Users typically use acronyms and slang to limit the characters in a tweet posted on social media [29]. The use of acronyms and slangis an important issue because the users do not have the same mindset to make the acronym in the same full form, and everyone considers the tweet in different styles or slang. Sometimes, the acronym posted may possess other meanings or be associated with other problems. So, interpreting these kinds of acronyms and replacing them with meaningful words should be done so the machine can easily understand the acronym's meaning.

An example illustrates the original tweet with acronyms and slang before and after removal.

Before removal: ROM permanently stores information in the system, whereas RAM temporarily stores information in the system.

After removal: Read Only Memory permanently store information in the system, whereas Random Access Memory temporarily store information in the system.

Removal of numbers

Removal of numbers in the Twitter dataset is a process of deleting the occurrence of numbers between any words in a sentence [29].

An example illustrates the original tweet before and after removing numbers.

Before removal: My ink "My Way…No Regrets" Always Make Happiness Your #1 Priority.

After removal: My ink "My Way … No Regrets" Always Make Happiness Your # Priority.

Once removed, the tweet will no longer contain any numbers.

Punctuation and symbol removal

The punctuation and symbols are removed in this stage. Punctuations such as ‘’, ‘,’, ‘?’, ‘!’, and ‘:’ are removed from the tweet [29, 30].

An example illustrates the original tweet before and after removing punctuation marks.

Before removal: My ink “My Way…No Regrets” Always Make Happiness Your #1 Priority.

After removal: My ink My Way No Regrets Always Make Happiness Your Priority.

After removal, the tweet will not contain any punctuation. Symbol removal is the process of removing all the symbols from the tweet.

An example illustrates the original tweet before and after removing symbols.

Before removal: wednesday addams as a disney princess keeping it [(100)] 👦💜♊□□🌵🌃🌙.

After removal: wednesday addams as a disney princess keeping it.

After removal, there would not be any symbols in the tweet.

Removal of uppercase into lowercase character

In this process of removal or deletion, all the uppercase charactersare replaced with lowercase characters [30].

An example illustrates the original tweet before and after removing uppercase characters into lowercase characters.

Before removal: My ink “My Way…No Regrets” Always Make Happiness Your #1 Priority.

After removal: my ink my way no regrets always make happiness your priority.

After removal, the tweet will no longer contain capital letters.

URL, hashtag & user mention removal

For clear reference,Twitter users post tweets with various URLs and hashtags [29, 30]. This information ishelpful for the people but mostly noise, which cannot be used for further processes. The example provided below illustrates the original tweet with URL, hashtag and user mention before removal and after removal:

Before removal: This gift is given by #ahearttouchingpersonfor securing @firstrank. Click on the below linkto know more https://tinyurl.com/giftvoucher.

After removal: This is a gift given by a heart touching person for securing first rank. Click on the below link to know more.

Term weighting-based feature extraction

After the pre-processing, the pre-processed data is extracted in text documents based on the term weighting T_w [22]. A new term weighting scheme,Log term frequency-based modified inverse class frequency (LTF-MICF), is employed in this research paper for feature extraction based on term weight. The technique integrates two different term weighting schemes: log term frequency (LTF) and modified inverse class frequency (MICF). The frequently occurring terms in the document are known as term frequency ^fT. But, ^fT alone is insufficient because the frequently occurring terms will possess heavyweight in the document. So, the proposed hybrid feature extraction technique can overcome this issue. Therefore, ^fT is integrated with MICF, an effective T_w approach. Inverse class frequency ^fC_i is the inverse ratio of the total class of terms that occurs on training tweets to the total classes. The algorithm for the TW-FE technique is shown in algorithm 1 [22].

```
Algorithm 1: LTF-MICF based $T_w$
Input: p – Pre-processed dataset
Output: Weight ranks for each term in the pre-processed dataset
Begin
For each term $tp\ (p = 1,2,.....m)$ and each class, $C_r\ (r = 1,2,......m)$ do
Compute $MICF(tp), w_{sp}.\lfloor {}^{f}C_i(tp) \rfloor$
Compute $w_{sp}$ and ${}^{f}C_i(tp)$ using equations (3) and (5)
end for
For each term, $tp\ (p = 1,2,....m)$ do
Compute ${}^{l}T_f(tp)$ using equation (4)
Compute $LTF - MICF\ (tp)$ using equation (2)
end for
Return $LTF - MICF\ (tp)(p = 1,2,.....m)$
End
```

Two steps are involved in calculating LTF ${}^{l}T_f$. The first step is to calculate the ${}^{f}T$ of each term in the pre-processed dataset. The second step is, applying log normalization to the output of the computed ${}^{f}T$ data. The modified version of ${}^{f}C_i$, the MICF is calculated for each term in the document. MICF is said to be executed then;each term in the document should have different class-particular ranks, which should possess differing contributions to the total term rank. It is necessary to assign dissimilar weights for dissimilar class-specific ranks. Consequently, the sum of the weights of all class-specific ranks is employed as the total term rank. The proposed formula for T_w using LTF-based MICF is represented as follows [22];

$$LTF - MICF(tp) =^{l} T_f(tp) * \sum_{r=1}^{m} w_{sp} - \left[{}^{f}C_i(tp)\right] \tag{2}$$

where a specific weighting factor is denoted w_{sp} for each tp for class C_r, which can be clearly represented as;

$$w_{sp} = \log(1 + \frac{s_i\ \overset{\rightharpoonup}{t}}{\max(1, s_i\ \overset{\leftarrow}{t}}.\frac{s_i\tilde{t}}{\max(1, s_i\widehat{t}} \tag{3}$$

The method used to assign a weight for a given dataset is known as the weighting factor (WF). Where the number of tweets s_i in class C_r which contains pre-processed terms tp is denoted as $s_i\ \overset{\rightharpoonup}{t}$. The number of s_i in other classes, which contains tp is denoted as $s_i\ \overset{\leftarrow}{t}$. The number of s_i in-class C_r, which do not possess,tp is denoted as $s_i\widehat{t}$. The number of s_i in other classes, which do not possess,tp is denoted as $s_i\tilde{t}$. To eliminate negative weights, the constant '1' is used. In extreme cases, to avoid a zero-denominator issue, the minimal denominator is set to '1' if $s_i\ \overset{\leftarrow}{t} = 0$ or $s_i\widehat{t} = 0$. The formula for ${}^{l}T_f(tp)$ and ${}^{f}C_i(tp)$ can be presented as follows [22];

$${}^{l}T_f(tp) = \log(1 +^{f} T(tp, s_i)) \tag{4}$$

where raw count of tp on s_i is denoted as ${}^{f}T(tp, s_i)$, i.e., the total times of tp occurs on s_i.

$$^{f}C_i(tp) = \log(1 + \frac{r}{C(tp)} \tag{5}$$

where r refers to the total number of classes in s_i, and $C(tp)$ is the total number of classes in tp. The dataset features are represented as $f_j = \{f_1, f_2, ...f_3, ...f_m\}$ after T_w, where the number of weighted terms in the pre-processed dataset is denoted as $f_1, f_2, ..f_3, ..f_m$ respectively. The computed rank values of each term in the text document of tweets are used for performing the further process.

Feature selection

The existence of irrelevant features in the data can reduce the accuracy level of the classification process and make the model to learn those irrelevant features. This issue is termed as the optimization issue. This issue can be ignored only by taking optimal solutions from the processed dataset. Therefore, a feature selection algorithm named White shark optimizer with a hybrid mutation strategy is utilized to achieve a feature selection process.

White Shark Optimizer (WSO)

WSO is proposed based on the behaviour of the white shark while foraging [23]. Great white shark in the ocean catches prey by moving the waves and other features to catch prey kept deep in the ocean. Since the white shark catch prey based on three behaviours, namely: (1) the velocity of the shark in catching the prey, (2) searching for the best optimal food source, (3) the movement of other sharks toward the shark, which is near to the optimal food source. The initial white shark population is represented as;

$$W^p_q = lb_q + r \times (up_q - lb_q) \tag{6}$$

where W^p_q is the initial parameters of the p_{th} white shark in the q_{th} dimension. The upper and lower bounds in the q_{th} dimension are denoted as up_q and lb_q, respectively. Whereas r denotes a random number in the range [0, 1].

The white shark's velocity is to locate the prey based on the motion of the sea wave is represented as [23];

$$vl^p_{s+1} = \mu\left[vl^p_s + F_1(W_{gbest_s} - W^p_s) \times C_1 + F_2(W^{vl^p_s}_{best} - W^p_s) \times C_2\right] \tag{7}$$

where $s = 1, 2, ...m$ is the index of a white shark with a population size of m. The new velocity of p_{th} shark is denoted as vl^p_{s+1} in $(s+1)_{th}$ step. The initial speed of the p_{th} shark in the s_{th} step is denoted as vl^p_s. The global best position achieved by any p_{th} shark in s_{th} step is denoted as W_{gbest_s}. The initial position of the p_{th} shark in s_{th} step is denoted as W^p_s. The best position of the p_{th} shark and the index vector on attainingthe best position are denoted as $W^{vl^p_s}_{best}$ and vc^i. Where C_1 and C_2 in the equation is defined as the creation of uniform random numbers of the interval [1, 0]. F_1 and F_2 are the force of the shark to control the effect of W_{gbest_s} and $W^{vl^p_s}_{best}$ on $W^p_s.\mu$ represents to analyze the convergence factor of the shark. The index vector of the white shark is represented as;

$$vc = [t \times rand(1, t)] + 1 \tag{8}$$

where $rand(1, t)$ is a random numbers vector obtained with a uniform distribution in the interval [0, 1].The forces of the shark to control the effect are represented as follows;

$$F_1 = F_{\max} + (F_{\max} - F_{\min}) \times e^{-(4u/U)^2} \tag{9}$$

$$F_2 = F_{\min} + (F_{\max} - F_{\min}) \times e^{-(4u/U)^2} \tag{10}$$

The initial and maximum sum of the iteration is denoted as u and U, whereas the white shark's current and sub-ordinate velocities are denoted as $F_{\min}$ and $F_{\max}$. The convergence factor is represented as;

$$\mu = \frac{2}{\left|2 - \tau - \sqrt{\tau^2 - 4\tau}\right|} \tag{11}$$

where τ is defined as the acceleration coefficient. The strategy for updating the position of the white shark is represented as follows;

$$W_{s+1}^{p} = \begin{cases} W_s^p .\neg \oplus W_\circ + up\ .c + lo\ .d;\ rand < MV \\ W_s^p + vl_s^p / fr;\ rand \geq MV \end{cases} \tag{12}$$

The new position of the p_{th} shark in $(s + 1)$ iteration, ¬ represent the negation operator, c and d represents the binary vectors. The search space lower and upper bounds are denoted as lo and ub. W_0 and fr denotes the logical vector and frequency at which the shark moves. The binary and logic vectors are expressed as follows;

$$c = \operatorname{sgn}(W_s^p - up)\ >\ 0 \tag{13}$$

$$d = \operatorname{sgn}(W_s^p - 1)\ >\ 0 \tag{14}$$

$$W_o = \oplus(c, d) \tag{15}$$

The frequency at which the white shark moves is represented as;

$$fr = fr_{\min} + \frac{fr_{\max} - fr_{\min}}{fr_{\max} - fr_{\min}} \tag{16}$$

$fr_{\max}$ and $fr_{\min}$ represents the maximum and minimum frequency rates. The increase in force at each iteration is represented as;

$$MV = \frac{1}{(c_0 + e^{(s/2 - S)/c_1})} \tag{17}$$

where MV represents the weight of the terms in the document.

The best optimal solution is represented as;

$$W'^{p}_{s+1} = W_{gbest_s} + r_1 \vec{D}is_W \operatorname{sgn}(r_2 - 0.5) r_3 < Str_{sns} \tag{18}$$

where the position updation following the food source of p_{th} the white shark is denoted as W'^{p}_{s+1}. The $\operatorname{sgn}(r_2 - 0.5)$ produce 1 or −1 to modify the search direction. The food source and shark distance $\vec{Dis}_w$ and the strength of the white shark following other sharks close to the food source Str_{sns} is formulated as follows;

$$\vec{Dis}_W = \left|rand \times (W_{gbest_s} - W_s^p\right| \tag{19}$$

$$Str_{sns} = \left|1 - e^{(c_2 \times s/S)}\right| \tag{20}$$

The initial best optimal solutions are kept constant, and the position of other sharks is updated according to these two constant optimal solutions. The fish school behaviour of the sharks is formulated as follows;

$$W_{S+1}^p = \frac{W_s^p + W_{s+1}^{\prime p}}{2 \times rand} \tag{21}$$

The weight factor $^j we$ is represented as;

$$^p we = \frac{1}{m-1} * \left(\frac{\sum_{\Upsilon=1, \Upsilon \neq j}^{\rho} {}^q fit}{\sum_{\Upsilon=1}^{\rho} {}^q fit} \right) \tag{22}$$

where $^q fit$ is defined as the fitness of each term in the text document. The expansion of the equation is represented as;

$$^p we = \frac{1}{m-1} * \frac{\left[{}^1 fit +^2 fit + ... +^{q+1} fit + ... +^{m-1} fit +^m fit\right]}{{}^1 fit +^2 fit + ...^{q-1} fit +^q fit +^{q+1} fit + ... +^{m-1} fit +^m fit} \tag{23}$$

The concatenation of hybrid mutation HM is applied to the WSO for a faster convergence process. Thus, the hybrid mutation applied with the optimizer is represented as;

$$HM =^{t+1} GM +^{t+1} CM \tag{24}$$

$$^{t+1} GM = W_q^{new} + D_1.G_a(\mu, \sigma) \tag{25}$$

$$^{t+1} CM = W_q^{new} + D_2.C_a(\mu', \sigma') \tag{26}$$

whereas $G_a(\mu, \sigma)$ and $C_a(\mu, \sigma)$ represents an arbitrary number of both Gaussian and Cauchy distribution. (μ, σ) and (μ', σ') represents the mean and variance function of both Gaussian and Cauchy distributions. D_1 and D_2 represents the coefficients of Gaussian ^{t+1}GM along with Cauchy ^{t+1}CM mutation. On applying these two hybrid mutation operators, a new solution is produced that is represented as;

$$[W_q^{new}]_{new} = W_q^{new} +^p we(HM) \tag{27}$$

where,

$$p_{we} = \sum\nolimits_{y=1}^{PS} \frac{W_q^{new}}{PS} \tag{28}$$

whereas ${}^{p}_{we}$ represents the weight vector and PS represents the size of the population. The selected features from the extracted features are represented as $Sel(p = 1, 2, ...m)$. The WSO output is denoted as $(sel) = \left\{sel^1, sel^2, ...sel^m\right\}$,which is a new sub-group of terms in the dataset. At the same time,m denotes a new number of each identical feature. Finally, the feature selection stage provides a dataset document with optimal features.

Gated attention recurrent network (GARN) classifier

GARN is a hybrid network of Bi-GRU with an attention mechanism. Many problems occur due to the utilization of recurrent neural network (RNNs) because it employs old information rather than the current information for classification. To overcome this problem, a bidirectional recurrent neural network (BRNN) model is proposed, which can utilize both old and current information. So, to perform both the forward and reverse functions, two RNNs are employed. The output will be connected to a similar output layer to record the feature sequence. Based on the BRNN model, another bidirectional gated recurrent unit (Bi-GRU) model is introduced, which replaces the hidden layer of the BRNN with a single GRU memory unit. Here, the hybridization of both Bi-GRU with attention is considered agated attention recurrent network (GARN) [25] and its structure is given in Fig. 2.

Consider an m-dimensional input data as $(y_1, y_2, ..., y_m)$. The hidden layer in the BGRU produces an output H_{t_1} at a time interval t_1 is represented as;

$$\vec{H}_{t_1} = \sigma\left(w_{e_{\underset{yH}{\longrightarrow}}} y_{t_1} + w_{e_{\underset{H}{\longrightarrow}\underset{H}{\longrightarrow}}} \vec{H}_{t_1-1} + c_{\underset{H}{\longrightarrow}}\right) \tag{29}$$

$$\overleftarrow{H}_{t_1} = \sigma\left(w_{e_{\underset{yH}{\longleftarrow}}} y_{t_1} + w_{e_{\underset{H}{\longleftarrow}\underset{H}{\longleftarrow}}} \overleftarrow{H}_{t_1-1} + c_{\underset{H}{\longleftarrow}}\right) \tag{30}$$

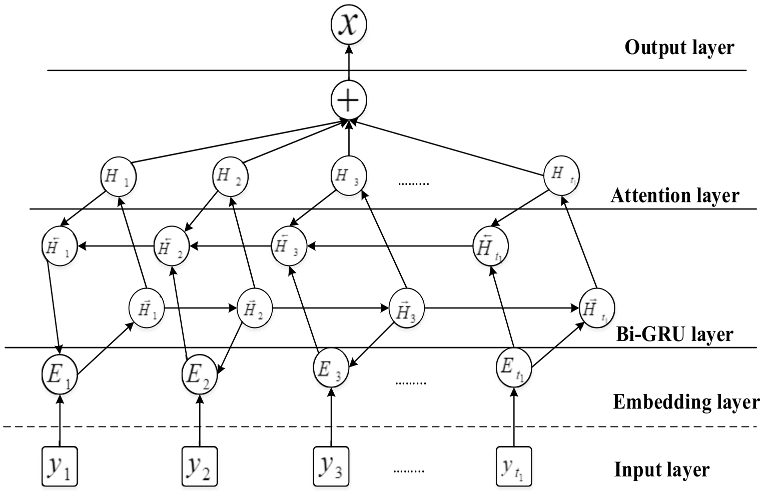


Fig. 2 Structure of GARN

$$H_{t_1} = \vec{H}_{t_1} \oplus \overleftarrow{H}_{t_1} \tag{31}$$

where the weight factor for two connecting layers is denoted as w_e, c is the bias vector, σ represents the activation function, positive and negative outputs of GRU is denoted as $\vec{H}_{t_1}$ and $\overleftarrow{H}_{t_1}$, $\oplus$ is a bitwise operator.

Attention mechanism

In sentiment analysis, the attention module is very important to denote the correlation between the terms in a sentence and the output [26]. For direct simplification, an attention model is used in this proposal named as feed-forward attention model. This simplification is to produce a single vector ν from the total sequence represented as;

$$E_{t_1} = b(H_{t_1}) \tag{32}$$

$$\beta_{t_1} = \frac{\exp(E_{t_1})}{\sum_{s=1}^{R} \exp(E_s)} \tag{33}$$

$$\nu = \sum_{t_1=1}^{R} \beta_{t_1} H_{t_1} \tag{34}$$

Where β is a learning function and is identified using H_{t_1}. From the above Eq. 34, the attention mechanism produces a fixed length for the embedding layer in a BGRU model for every single vector ν by measuring the average weight of the data sequence H. The structure for attention mechanism is shown in Fig. 3. Therefore, the final sub-set for the classification is obtained from:

$$H^{\#} = \tanh(\nu) \tag{35}$$

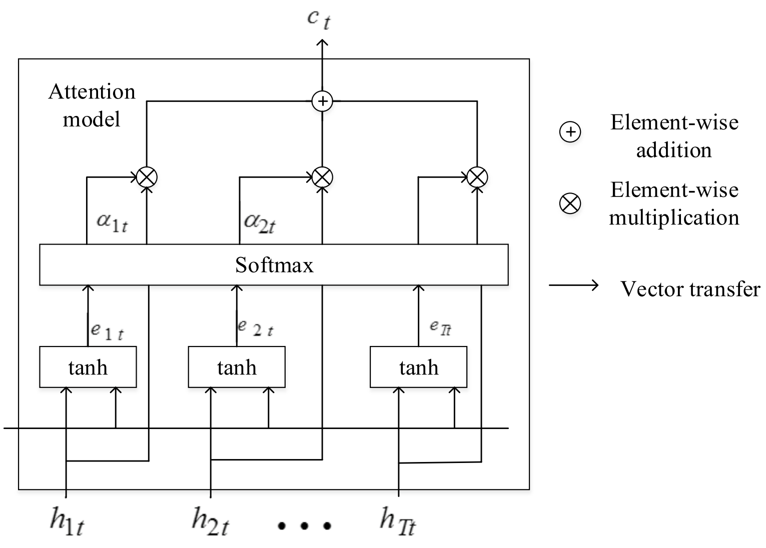


Fig. 3 Structure of attention mechanism

Sentiment classification

Twitter sentiment analysis is formally a classification problem. The proposed approach classifies the sentiment data into three classes: positive, negative and neutral. For classification, the softmax classifier is used to classify the output in the hidden layer $H^{\#}$ is represented as;

$$Q(x|T) = soft\max\left(w_e H^{\#} + c\right) \tag{36}$$

$$x = \arg\max_{x} Q(x|T) \tag{37}$$

where w_e is the weight factor, c is a bias vector and $H^{\#}$ is the output of the last hidden layer. Also, the cross-entropy is evaluated as a loss function represented as;

$$loss_{fc} = -\frac{1}{n}\sum_{j=1}^{n} sen_j \log x_j + \lambda||\theta||^2 \tag{38}$$

The total number of samples is denoted as, n. The real category of the sentence is denoted as sen_j,the sentence with the predictive category is denoted as x_j, and the $L2$ regular item is denoted as $\lambda||\theta||^2$.

Results and discussion

This section briefly describes the performance metrics like accuracy, precision, recall and f-measure. The overall analysis of the Twitter sentiment classification with preprocessing, feature extraction, feature selection and classification are also analyzed and discussed clearly. Results on comparing the existing and trending classifiers with term weighting schemes in bar graphs and tables are included. Finally, a small discussion about the overall workflow concluded the research by importing the analyzed performance metrics. The sentiment is an expression from individuals based on an opinion on any subject. Tweet-based analysis of sentiment mainly focuses on detecting positive and negative sentiments. So, it is necessary to enhance the classification classes in which a neutral class is added to the datasets.

Dataset

The dataset utilized in our proposed work is Sentiment 140, gathered from [27], which contains 1,600,000tweets extracted from Twitter API. The score values for each tweet as, for positive tweets, the rank value is 4.Similarly, for negative tweets rank value is 0, and for neutral tweets, the rank value is 2.The total number of positive tweets in a dataset is 20832, neutral tweets are 18318, negative tweets are 22542, and irrelevant tweets are 12990. From the entire dataset, 70%is used for training, 15% for testing and 15% for validation. Table 2 shows the system configuration of the designed classifier.

Performance metrics

In this proposed method, 4 different weight schemes are compared with other existing,proposed classifiers in which the performance metrics are precision, f1-score, recall and accuracy. Four notations, namely, true-positive (t_p), true-negative (t_n),

Table 2 System configuration of the designed classifier

Serial No	Parameters	Configuration
1	Device name	DESKTOP-NDIBIU4.smg.local
2	Processor	Intel (R) Core (TM) i5-6500 CPU @ 3.20 GHz, 3.19 GHz
3	Installed RAM	16.0 GB (15.9 GB unstable)
4	Device ID	6C4646EC-BA2C-4DC1-AA1F-6F0E989718EF
5	Product ID	00,342–50,603-46,281-AAOEM
6	System type	64-bit operating system, × 64-based processor
7	Pen and touch	No pen or touch input is available for this display

false-positive (f_p) and false-negative, (f_n) are particularly utilized to measure the performance metrics.

Accuracy (A_c)

Accuracy is the dataset's information accurately being classified by the proposed classifier. The accuracy value for the proposed method is obtained using Eq. 39.

$$A_c = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{39}$$

Precision (P_r)

Precision is defined as the number of terms accurately identified positive to the total identified positively. The precision value for the proposed method is obtained using Eq. 40.

$$P_r = \frac{t_p}{t_p + f_p} \tag{40}$$

Recall (R_e)

The recall is defined as the percentage of accurately identified positive observations to the total observations in the dataset. The recall value for the proposed method is obtained using Eq. 41.

$$R_e = \frac{t_p}{t_p + f_n} \tag{41}$$

F1-score (F_s)

F1-score is defined as the average weight of recall and precision. The f1-score value for the proposed method is obtained using Eq. 42.

$$F_s = 2\frac{P_r . R_e}{P_r + R_e} \tag{42}$$

Analysis of Twitter sentiments using GARN

The research paper mainly focuses on classifying Twitter sentiments in the form of three classes, namely, positive, negative and neutral. The data are collected using Twitter api. After collecting data, it is given as input for pre-processing. The unwanted symbols are removed in the pre-processing technique, giving a new pre-processed dataset. Now, the pre-processed dataset is given as an input to extract the required features. These features are extracted from the pre-processed dataset using a novel technique known as the log term frequency-based modified inverse class frequency (LTF-MICF) model, which integrates two-weight schemes, LTF and MICF. Here, the required features are extracted in which the extracted features are given as input to select an optimal feature subset. The optimized feature subset is selected using a hybrid mutation-based white shark optimizer (HMWSO). The mutation is referred to as the Cauchy mutation and the Gaussian mutation. Finally, with the selected feature sub-set as input, the sentiments are classified under three classes using a classifier named gated recurrent attention recurrent network (GARN), which is a hybridization of Bi-GRU with an attention mechanism.

The evaluated value of the proposed GARN is preferred for classifying the sentiments of Twitter tweets. The suggested GARN model is implemented in the Python environment, and the sentiment140 Twitter dataset is utilized for training the proposed model. To evaluate the efficiency of the classifier, the proposed classifier is compared with existing classifiers, namely, CNN (Convolutional neural network), DBN (Deep brief neural network), RNN (Recurrent neural network), and Bi-LSTM (Bi-directional long short term memory). Along with these classifiers, the proposed term weighting scheme (LTF-MICF) with the existing term weighting schemes TF (Term Frequency), TF-IDF (Term-frequency-inverse document frequency), TF-DFS (Term-frequency-distinguishing feature selector), and W2V (Word to vector) are also analyzed. The performance was evaluated for both sentiment classification with an optimizer and without using an optimizer. The metrics evaluated are accuracy, precision, recall and f1-score, respectively. The existing methods implemented and proposed (GARU) are Bi-GRU, RNN, Bi-LSTM, and CNN. The simulation parameters used for processing the proposed and existing methods are discussed in Table 3. This comparative analysis is performed to show the efficiency of a proposed over the other related existing algorithms.

Figure 4 compares the accuracy of the GARN with the existing classifiers. The accuracy obtained by existing Bi-GRU, Bi-LSTM, RNN, and CNN for the LTF-MICF is 96.93%, 95.79%, 94.59% and 91.79%. In contrast, the proposed GARN classifier achieves an accuracy of 97.86% and is considered the best classifier with the LTF-MICF term weight scheme for classifyingTwitter sentiments. But when the proposed classifier is compared with other term weighting schemes,TF-DFS, TF-IDF, TF and W2V, the accuracy obtained is 97.53%, 97.26%, 96.73% and 96.12%. Therefore, the term weight scheme withthe GARN classifier is the best solution for classification problems. Table 4 contains the accuracy values attained by four existing classifiers and the proposed classifier with four existing term weight schemes and proposed term weight scheme.

Figure 5 shows the precision performance analysis with the proposed and four existing classifiers for different term weight schemes. The precision of all existing classifiers with other term weight schemes is less than the proposed term weighting scheme. In Bi-GRU, the precision obtained by TF-DFS, TF-IDF, TF and W2V is 94.51%,

Table 3 Simulation parameters of proposed and existing implemented methods

Method	Layers	Value
GARU (proposed)	Bi_directional GRU	500
	Bi_directional GRU	250
	Attention layer	size of Bi_GRU
	Dropout	0.2
	Dense	100
	Dropout	0.2
	Dense	1
Bi_GRU	Bi_directional GRU	500
	Bi_directional GRU	200
	Dropout	0.2
	Dense	100
	Dropout	0.2
	Dense	1
RNN	Embedding layer	input_dim = 100
	GRU layer	256
	Simple RNN layer	128
	Dense	3
Bi_LSTM	Embedding layer	top_words = 10, n = 128
	Bidirectional LSTM layer	64
	Dropout	0.5
	Dense	1
CNN	Input layer	69,769*1000
	Convolution layer	Filter = 2, Kernel size = 2
	Maxpooling layer	Pool size = 2
	Flatten	Size of maxpool
	Dense layer	1

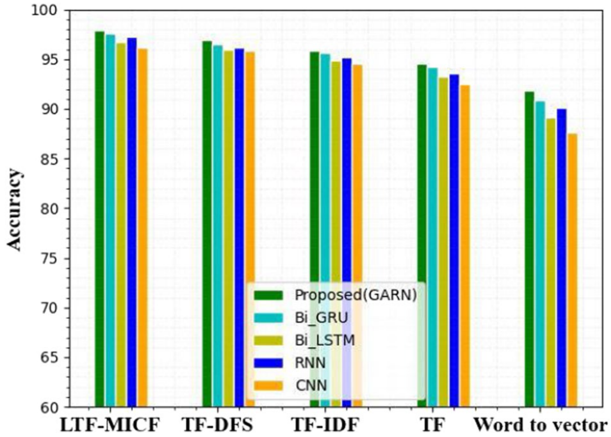


Fig. 4 Accuracy of the classifiers with term weight schemes

94.12%, 93.76% and 93.59%. But, when Bi-GRU is compared with the LTF-MICF term weight scheme, the precision level is increased by 95.22%. The precision achieved by the suggested method GARN with TF-DFS, TF-IDF, TF and W2V is 96.03%, 95.67%, 94.90% and 93.90%. Whereas, when the GARN classifier is compared with the suggested term weighting scheme LTF-MICF the precision achieved is 96.65%, which is

Table 4 Accuracy of the proposed and existing classifiers with term weights

Classifiers	Term weighting-based feature extraction schemes				
	LTF-MICF %	TF-DFS %	TF-IDF %	TF %	W2V %
Proposed GARN	97.86	97.53	97.26	96.73	96.12
Bi-GRU	96.93	96.46	96.19	95.92	95.79
Bi-LSTM	95.79	95.59	95.19	94.86	94.59
RNN	94.59	94.19	93.52	93.25	92.45
CNN	91.79	90.85	90.05	89.12	87.65

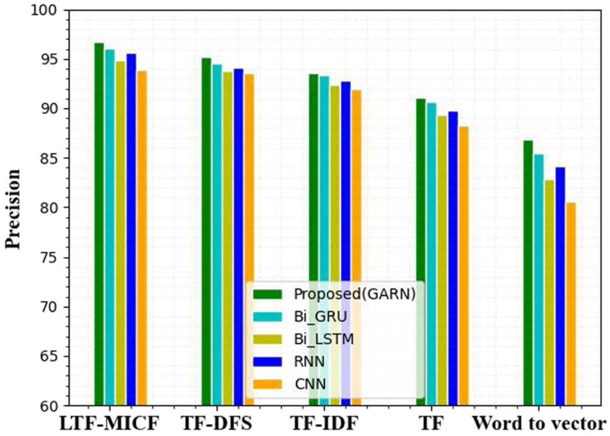


Fig. 5 Precision of the classifiers with term weight schemes

Table 5 Precision of the proposed and existing classifiers with term weights

Classifiers	Term weighting-based feature extraction schemes				
	LTF-MICF %	TF-DFS %	TF-IDF %	TF %	W2V %
Proposed GARN	96.65	96.03	95.67	94.90	93.90
Bi-GRU	95.22	94.51	94.12	93.76	93.59
Bi-LSTM	93.61	93.34	92.82	92.37	91.99
RNN	91.11	90.65	89.74	89.38	88.28
CNN	86.83	85.41	84.17	82.84	80.63

considered the best classifier with the best term weighting scheme. Figure 5 shows that the GARN classifier with the LTF-MICF term weighting scheme achieved the highest precision level compared with other classifiers and term weighting schemes. Table 5 indicates the precision performance analysis for existing and proposed classifiers with term weight schemes.

The analysis graph of Fig. 6 shows the f-measure of the four prevalent classifiers and suggested classifiers with different term weight schemes. The f-measure of all the prevalent classifier with other term weight schemes are minimum compared to the suggested term weighting scheme. In Bi-LSTM, the f-measure gained with TF-DFS, TF-IDF, TF and W2V is93.34%, 92.77%, 92.28% and 91.89%. Compared with LTF-MICF, the

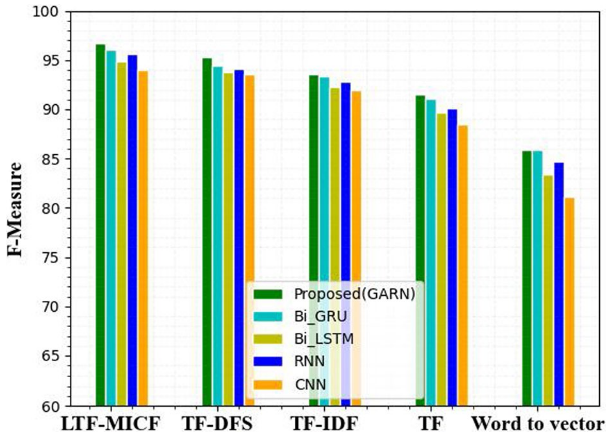


Fig. 6 F-measure of the classifiers with term weight schemes

Table 6 F-measure of the proposed and existing classifiers with term weights

Classifiers	Term weighting-based feature extraction schemes				
	LTF-MICF %	TF-DFS %	TF-IDF %	TF %	W2V %
Proposed GARN	96.70	96.10	95.65	94.90	94.00
Bi-GRU	95.27	94.48	94.09	93.73	93.52
Bi-LSTM	93.61	93.34	92.77	92.28	91.89
RNN	91.52	91.05	90.05	89.69	88.50
CNN	87.30	85.92	84.68	83.35	81.10

f-measure level is improved by 95.22%. The f-measure derived by the advance GARN with TF-DFS, TF-IDF, TF and W2V is 96.10%, 95.65%, 94.90% and 94.00%. When GARN is compared with the advanced LTF-MICF scheme, the f-measure grows by 96.70%, which is considered the leading classifier with the supreme term weighting scheme. Therefore, from Fig. 6, the GARN model with the LTF-MICF scheme achieved the greatest f-measure level compared with other DL models and term weighting schemes.Table 6 indicates the performance analysis of the f-measure for both prevalent and suggested classifiers with term weight schemes.

Figure 7 illustrates the recall of the four previously discovered DL models andthe recommended model of dissimilar term weight schemes. The recall of the previously discovered classifier with other term weight schemes is reduced compared to the novel term weighting scheme. In RNN, the recall procured with TF-DFS, TF-IDF, TF and W2V is 91.83%, 90.65%, 90.36% and 89.04%. In comparison with LTF-MICF, the recall value is raised by 92.25%. The recall acquired by the invented GARN with TF-DFS, TF-IDF, TF and W2V is 96.23%, 95.77%, 94.09% and 94.34%. Comparing GARN with the advanced LTF-MICF scheme maximizes recall by 96.76%,which is appraised as the prime classifier with an eminent term weighting scheme. Therefore, from Fig. 7, the GARN model with the LTF-MICF scheme securedextraordinaryrecallvalue when differentiated from other DL models and term weighting schemes. Table 7 indicates the recall performance analysis for the previously discovered and recommended classifiers with term weight schemes.

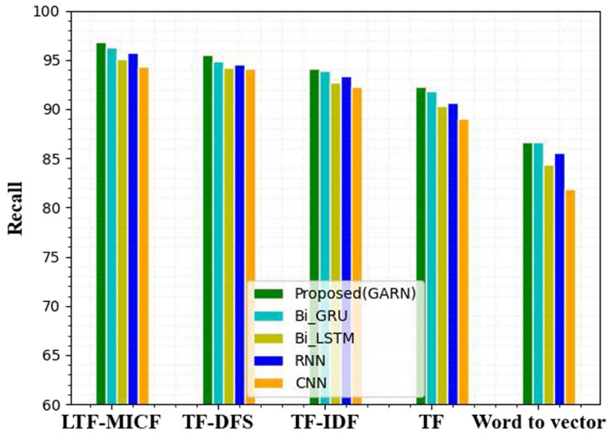


Fig. 7 Recall of the classifiers with term weight schemes

Table 7 Recall of the proposed and existing classifiers with term weights

Classifiers	Term weighting-based feature extraction schemes				
	LTF-MICF %	TF-DFS %	TF-IDF %	TF %	W2V %
Proposed GARN	96.76	96.23	95.77	95.09	94.34
Bi-GRU	95.51	94.84	94.55	94.25	94.06
Bi-LSTM	94.07	93.85	93.36	92.75	92.27
RNN	92.25	91.83	90.65	90.36	89.04
CNN	87.96	86.64	85.52	84.41	81.88

Discussion

The four stages employed to implement this proposed work are Twitter data collection, tweet pre-processing, term weighting-based feature extraction, feature selection and classification of sentiments present in the tweet. Initially, the considered tweet sentiment dataset is subjected to pre-processing.Here, tokenization, stemming, punctuations, symbols, numbers, hashtags, and acronyms are removed. After removal, a clean pre-processed dataset is obtained. The performance achieved by proposed and existing methods for solving proposed objective is discussed in Table 8.

Using this pre-processed dataset, a term weighting-based feature extraction is done using an integrated terms weight scheme such as LTF and MICF as a novel term weighting scheme technique named LTF-MICF technique. An optimization algorithm, HMWSO, with two hybrid mutation techniques, namely Cauchy and Gaussian mutation, is chosen for feature selection. Finally, the GARN classifier is used for the classification of Twitter sentiments. The sentiments are classified as positive, negative and neutral. The performance of existing classifiers with term weighting schemes and the proposed classifier with term weighting schemes are analyzed. The performance comparison between the proposed and existing methods is shown in Table 9. The existing details are collected from previous works developed for sentiment analysis from theTwitter dataset.

Table 8 Performance between proposed and existing methods for developed objective

Models	Accuracy	Precision	Sensitivity	F1-measure
Proposed (GARN)	97.86453	96.6588	96.764	96.7
BiLSTM	95.7957	93.6148	94.07	93.613
BiGRU (Bidirectional gated recurrent unit)	95.1951	92.82	93.36	92.7
TCN (Temporal convolutional network)	95.9302	94.221	94.519	94.275
Transformer	96.007	95.8	95.76	94.89
BERT (Bidirectional Encoder Representations from Transformers)	95	94.32	94.67	93.92

Table 9 Performance comparison between developed existing and proposed methods

Ref no and author name	Technique	Dataset	Performance metric
Proposed	GARN	Sentiment140 dataset	Accuracy–97.86%
Alharbi et al. 2019 [16]	CNN	SemEval-2016 1, and SemEval-2016 2	Accuracy–86.48%, precision–88%, recall–89%, and F1-score–87%
Tam et al. 2021 [17]	ConvBiLSTM	Retrieved Tweets and SST-2 datasets	Accuracy–91.13%, precision–94.6%, recall–94.33%, and F1-score–92.08%
Chugh et al. 2021 [18]	DeepRNN-SMCA	Amazon unlocked the mobile reviews dataset, Telecom tweets	Accuracy–97.7%, precision–95.5%, recall–94.6%, and F1-score–96.7%
Alamoudi et al. 2021 [19]	ALBERT	Yelp Dataset	Accuracy–89.49%, precision–89.02%, recall–89.49%, and F1-score–89.21%
Tan et al. 2022 [20]	RoBERTa-LSTM	Sentiment 140 dataset	Accuracy–89.7%, precision–90%, recall–90%, and F1-score–90%
Hasib et al. 2021 [21]	DNN-CNN	CrowdFlower Twitter US Airline Sentiment	Accuracy–91%, precision–85.66%, recall–87.33%, and F1-score–87.66%
Gaye, B et al., 2021 [31]	LR-LSTM	Sentiment 140 dataset	Accuracy–80%, precision–81%, recall–80%, and F1-score–90%
Ahmed, K et al., 2022 [32]	GA(SAE)-SVM	Sentiment 140 dataset	Accuracy–84.5%, precision–84.2%, recall–83.6%, and F1-score–83.9%
Subba, B. and Kumari, S, 2022 [33]	Bi-GRU-LSTM	Sentiment 140 dataset	Accuracy–84%, precision–85%, recall–83%, and F1-score–84%

Many DL techniques use only a single feature extraction technique, namely term frequency (TF) and distinguishing feature selector (DFS), which will not accurately extract the features. The proposed methods without optimization can diminish the proposed model's accuracy level. The feature extraction technique used in our proposed work will perform greatly because it can extract features from frequently occurring terms in the document. The proposed work uses an optimization algorithm to increase the accuracy level of the designed model.The achieved results are shown in Fig. 8.

The accuracy comparison by varying the total selected features is described in Fig. 9 (a). The ROC curve of proposed model is discussed in Fig. 9 (b). The ROC is evaluated using FPR (False positive rate), and TPR (True positive rate). The AUC (Area under curve) obtained for proposed is found to be 0.989. It illustrates that the proposed model has shown efficient accuracy with less error rate.

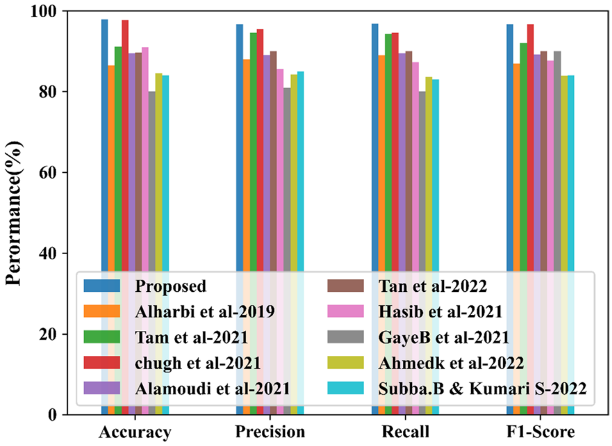
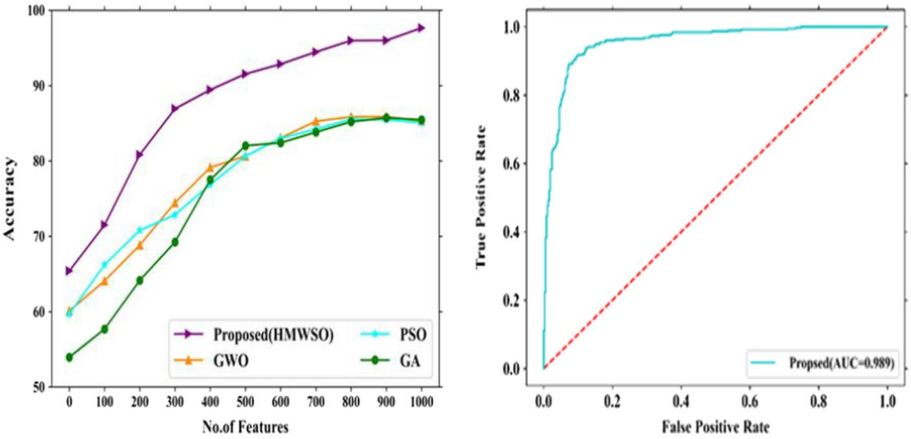


Fig. 8 Performance comparison between proposed and existing methods



a)

b)

Fig. 9 **a** Accuracy vs no of features **b** ROC curve

Ablation study

The ablation study for the proposed model is discussed in Table 10. In this the performance of overall architecture is described, further the comparative analysis between existing techniques also described in Table 10. Among all the techniques the proposed GARN has attained efficient performance than other algorithms. The hybridized methods are separately analysed and the results achieved by such techniques are also analysed which indicates that the integrating of all methods have improved the overall efficiency than applying the techniques in separate manner. Along with that, the ablation study for feature selection process is also evaluated and the obtained results are provided in Table 10.The existing classification and feature selection methods taken for comparison are GRN (Gated recurrent network), ARN (Attention based recurrent network), RNN (Recurrent neural network), WSO, and MO (Mutation optimization).

The computational complexity of proposed model is defined below:The complexity of attention model is $O\left(n^2 \cdot d\right)$, for recurrent network it is $O\left(n \cdot d^2\right)$, and the complexity of gated recurrent is found to be $O\left(k \cdot n \cdot d^2\right)$. The total complexity of proposed GARN is

Table 10 Result analysis related to ablation experiment

Methods	Models	Accuracy	Precision	Recall	F1-score
Classification	GARN (Proposed)	97.864	96.658	96.764	96.706
	GRN	96.104	95.67	95.77	94.86
	ARN	95.67	94.904	94.34	92.75
	RNN	93.19	94	93.9	91.5207
Feature selection	HMWSO	97.864	95.99	96.673	95.897
	WSO	96.104	96.750	95.9	94.35
	MO	95	94.3	94.2	95.7

$O\left(k \cdot n^2 \cdot d\right)$. This complexity shows that the proposed model has obtained efficient performance by reducing the system complexity. However, using the model separately won't provide satisfactory performance. However, integration of such models has attained efficient performance than other existing methods.

Conclusion

GARN is preferred in this research to find the various opinions of Twitter online platform users. The implementation was carried out by utilizing the Sentiment 140 dataset. The performance of the leading GARN classifier is compared with other DL models Bi-GRU, Bi-LSTM, RNN and CNN for four performance metrics: accuracy, precision, f-measure and recall centred with four-term weighting schemes LTF-MICF, TF-DFS, TF-IDF, TF and W2V. The evaluation shows that the leading GARN DL technique reached the target level for Twitter sentiment classification. Additionally, while applying the suggested term weighting scheme-based feature extraction technique LTF-MICF with the leading GARN classifier gained an efficient result for tweet feature extraction. With the Twitter dataset, the GARN accuracy on applying LTF-MICF is 97.86%. The accuracy value attained by the proposed classifier is the highest of all the existing classifiers. Finally, the suggested GARN classifier is regarded as an effective DL classifier for Twitter sentiment analysis and other sentiment analysis applications. The proposed model has attained satisfactory result but it haven't attained required level. This is because the proposed architecture fails to provide equal importance to the selected features. Due to this, few of the important features get lost, this has reduced the efficient performance of proposed model.Therefore as a future scope, an effective DL technique with the best feature selection method for classifying visual sentiment classification by utilizing all the selected features will be introduced. Further, this method is analysed using the small dataset, therefore in future large data with challenging images will be used to analyse the performance of present architecture.

Abbreviations

DL	Deep Learning
GRAN	Gated recurrent attention network
LTF-MICF	Log Term Frequency-based Modified Inverse Class Frequency
HMWSO	Hybrid mutation based white shark optimizer
RNN	Recurrent neural network
NLP	Natural Language Processing
SVM	Support Vector Machine
NB	Naïve Bayes

TSA	Twitter Sentiment Analysis
CNN	Convolutional Neural Network
TBRS	Term based random sampling

Acknowledgements
Not applicable.

Author contributions
NP and PC has found the proposed algorithms and obtained the datasets for the research and explored different methods discussed and contributed to the modification of study objectives and framework. Their rich experience was instrumental in improving our work. BTH and AS has done the literature survey of the paper and contributed writing the paper. All authors contributed to the editing and proofreading. All authors read and approved the final manuscript.

Funding
Authors did not receive any funding for this study.

Availability of data and materials
In this work, the dataset utilized in our proposed work contains 1,600,000 with score values for each tweets as, for positive tweets the rank value is 4 similarly for negative tweets rank value is 0 and for neutral tweets the rank value is 2 are collected using twitter api.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no Competing interests.

Received: 30 June 2022 Accepted: 3 April 2023
Published online: 17 April 2023

References

1. Saberi B, Saad S. Sentiment analysis or opinion mining: a review. Int J Adv Sci Eng Inf Technol. 2017;7(5):1660–6.
2. Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: a survey. Ain Shams Eng J. 2014;5(4):1093–113.
3. Drus Z, Khalid H. Sentiment analysis in social media and its application: systematic literature review. Procedia Comput Sci. 2019;161:707–14.
4. Zeglen E, Rosendale J. Increasing online information retention: analyzing the effects. J Open Flex Distance Learn. 2018;22(1):22–33.
5. Qian Yu, Deng X, Ye Q, Ma B, Yuan H. On detecting business event from the headlines and leads of massive online news articles. Inf Process Manage. 2019;56(6): 102086.
6. Osatuyi B. Information sharing on social media sites. Comput Hum Behav. 2013;29(6):2622–31.
7. Neubaum, German. Monitoring and expressing opinions on social networking sites–Empirical investigations based on the spiral of silence theory. PhD diss., Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2016, 2016.
8. Karami A, Lundy M, Webb F, Dwivedi YK. Twitter and research: a systematic literature review through text mining. IEEE Access. 2020;8:67698–717.
9. Antonakaki D, Fragopoulou P, Ioannidis S. A survey of Twitter research: data model, graph structure, sentiment analysis and attacks. Expert Syst Appl. 2021;164: 114006.
10. Birjali M, Kasri M, Beni-Hssane A. A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl-Based Syst. 2021;226: 107134.
11. Yadav N, Kudale O, Rao A, Gupta S, Shitole A. Twitter sentiment analysis using supervised machine learning. Intelligent data communication technologies and internet of things. Singapore: Springer; 2021. p. 631–42.
12. Jain PK, Pamula R, Srivastava G. A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. Comput Sci Rev. 2021;41:100413.
13. Pandian AP. Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis. Journal of Soft Computing Paradigm (JSCP). 2021;3(02):123–34.
14. Gandhi UD, Kumar PM, Babu GC, Karthick G. Sentiment analysis on Twitter data by using convolutional neural network (CNN) and long short term memory (LSTM). Wirel Pers Commun. 2021;17:1–10.
15. Kaur H, Ahsaan SU, Alankar B, Chang V. A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets. Inf Syst Front. 2021;23(6):1417–29.
16. Alharbi AS, de Doncker E. Twitter sentiment analysis with a deep neural network: an enhanced approach using user behavioral information. Cogn Syst Res. 2019;54:50–61.
17. Tam S, Said RB, Özgür Tanriöver Ö. A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification. IEEE Access. 2021;9:41283–93.

18. Chugh A, Sharma VK, Kumar S, Nayyar A, Qureshi B, Bhatia MK, Jain C. Spider monkey crow optimization algorithm with deep learning for sentiment classification and information retrieval. IEEE Access. 2021;9:24249–62.
19. Alamoudi ES, Alghamdi NS. Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. J Decis Syst. 2021;30(2–3):259–81.
20. Tan KL, Lee CP, Anbananthen KSM, Lim KM. RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access. 2022;10:21517–25.
21. Hasib, Khan Md, Md Ahsan Habib, Nurul Akter Towhid, Md Imran Hossain Showrov. A Novel Deep Learning based Sentiment Analysis of Twitter Data for US Airline Service. In 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), pp. 450–455. IEEE. 2021.
22. Zhao H, Liu Z, Yao X, Yang Q. A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. Inf Process Manage. 2021;58(5): 102656.
23. Braik M, Hammouri A, Atwan J, Al-Betar MA, Awadallah MA. White Shark Optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowl-Based Syst. 2022;243: 108457.
24. Carvalho F. Guedes, GP. 2020. TF-IDFC-RF: a novel supervised term weighting scheme. arXiv preprint arXiv:2003.07193.
25. Zeng L, Ren W, Shan L. Attention-based bidirectional gated recurrent unit neural networks for well logs prediction and lithology identification. Neurocomputing. 2020;414:153–71.
26. Niu Z, Yu Z, Tang W, Wu Q, Reformat M. Wind power forecasting using attention-based gated recurrent unit network. Energy. 2020;196: 117081.
27. https://www.kaggle.com/datasets/kazanova/sentiment140
28. Ahuja R, Chug A, Kohli S, Gupta S, Ahuja P. The impact of features extraction on the sentiment analysis. Procedia Comput Sci. 2019;152:341–8.
29. Gupta B, Negi M, Vishwakarma K, Rawat G, Badhani P, Tech B. Study of Twitter sentiment analysis using machine learning algorithms on Python. Int J Comput Appl. 2017;165(9):29–34.
30. Ikram A, Kumar M, Munjal G. Twitter Sentiment Analysis using Machine Learning. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) pp. 629–634. IEEE. 2022.
31. Gaye B, Zhang D, Wulamu A. A Tweet sentiment classification approach using a hybrid stacked ensemble technique. Information. 2021;12(9):374.
32. Ahmed K, Nadeem MI, Li D, Zheng Z, Ghadi YY, Assam M, Mohamed HG. Exploiting stacked autoencoders for improved sentiment analysis. Appl Sci. 2022;12(23):12380.
33. Subba B, Kumari S. A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. Comput Intell. 2022;38(2):530–59.
34. Pu X, Yan G, Yu C, Mi X, Yu C. Sentiment analysis of online course evaluation based on a new ensemble deep learning mode: evidence from Chinese. Appl Sci. 2021;11(23):11313.
35. Chen J, Chen Y, He Y, Xu Y, Zhao S, Zhang Y. A classified feature representation three-way decision model for sentiment analysis. Appl Intell. 2022;1:1–13.
36. Jain DK, Boyapati P, Venkatesh J, Prakash M. An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. Inf Process Manage. 2022;59(1): 102758.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com