International Journal of Cognitive Computing in Engineering 5 (2024) 161–169

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/

Twitter sentiment analysis using conditional generative adversarial network

V. Mahalakshmi [a,*], P. Shenbagavalli [b], S. Raguvaran [c], V. Rajakumareswaran [d], E. Sivaraman [e]

[a] Department of Information Technology, Sengunthar College of Engineering Tiruchengode 637205, India
[b] Assistant Professor, Karunya Institute of Technology and Sciences, Coimbatore
[c] Assistant Professor, Department of Computational Intelligence, School of Computing, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kanchipuram, Chennai, Tamil Nadu 603203, India
[d] Assistant Professor, Department of Computer Science and Design Erode Sengunthar Engineering College, Thuduppathi, Tamil Nadu, India
[e] Department of Electrical and Computer Engineering Curtin University, Miri, Malaysia

ARTICLE INFO

Keywords:
Conditional deep learning
Convolutional neural networks
And generative adversarial networks
Social media
Textual data

ABSTRACT

Sentiment analysis, which aims to extract information from textual data indicating people's ideas or attitudes about a particular problem, has developed into one of the most exciting study issues in natural language processing (NLP) with the development of social media. Twitter is a social network with an extensive audience that expresses their thoughts and opinions clearly and readily. Due to the prevalence of slang phrases and incorrect spellings in short phrase styles, Twitter data analysis is more challenging than data analysis from other social networks. Automated feature selection still has several limitations, such as higher computing costs that rise with the number of characteristics. Deep learning, which is self-learned and more accurate at processing vast amounts of data, is utilized to overcome these challenges. This paper introduces a conditional generative adversarial network (GAN) for Twitter sentiment analysis, whereas a convolutional neural network (CNN) has been used to extract traits from Twitter data. Compared to existing works, the proposed work has outperformed in accuracy, recall, precision, and F1 score. The suggested method is the most accurate, with a classification accuracy of 93.33 %.

1. Introduction

The exponential growth and accessibility of the internet have encouraged people to share their thoughts or points of view on social networks. Nowadays, social media is used to spread most of the information about society (Ahmad et al., 2019). Digital living has attracted billions of users to social networks like Facebook, Twitter, Instagram, and LinkedIn. Users of Twitter, a well-known microblogging platform, post 140-character messages called tweets in real time to share their thoughts, views, and ideas (Aljameel et al., 2020). Researchers have looked at Twitter data for many objectives, such as sentiment analysis, public health monitoring, political tendencies, education, and sports (Al-Zoubi et al., 2021).

With the content of users' tweets like topics, people, issues, events, services, commodities, and organizations, Twitter sentiment analysis (TSA) aims to determine how users feel or perceive certain entities and their attributes. The success of Twitter has increased the popularity of sentiment analysis (TSA), which falls into three main categories: statistical techniques, lexicon-based approaches (knowledge-based methods), and hybrid approaches (Abdulsaheb et al., 2023). Scalable and effectively used computing resources are lexicon-based approaches. Meanwhile, the lexicon-based strategy could improve at detecting emotion when language norms are considered (Ayyub et al., 2020). To obtain accurate polarity classification results, most prior sentiment analysis techniques trained shallow models on carefully chosen beneficial properties. These models commonly use classic classification methods to categorize linguistic characteristics such as n-grams, part-of-speech (POS) tags, and linguistic features such as support vector machines (SVM), latent Dirichlet allocation (LDA), and Naive Bayes (Basiri et al., 2021). However, because feature engineering is so complicated, the consequences of these methods depend heavily on feature representation, making it challenging to get good classification results (Bommaraju et al., 2017). Personality analysis, OSN age group categorization, and sentiment analysis are just a few areas where the deep learning technique has been researched (Ce & Tie, 2020).

Since deep learning emerged as a machine learning subset, sentiment analysis has advanced enormously. Computer algorithms with several

* Corresponding author.
E-mail addresses: mahalakshmivm15@gmail.com (V. Mahalakshmi), shenbagavalli@karunya.edu (P. Shenbagavalli), prof.sraguvaran@gmail.com (S. Raguvaran), mailtoraja@gmail.com (V. Rajakumareswaran), sivaraman.eswaran@gmail.com (E. Sivaraman).

https://doi.org/10.1016/j.ijcce.2024.03.002
Received 24 December 2022; Received in revised form 6 March 2024; Accepted 6 March 2024
Available online 7 March 2024

processing layers can learn various data representations while accounting for multiple levels of abstraction due to deep learning. As (Chandra & Krishna, 2021) points out, deep learning algorithms employ many processing layers to extract significant properties from data without requiring human input. Word embedding is the most popular and successful word vector encoding method for maintaining semantic and grammatical information (Cui et al., 2018).

Deep learning systems require time-consuming and expensive labeled data acquisition, and generative networks are helpful in this situation. Owing to GAN, categorization accuracy has improved (Divyapushpalakshmi & Ramalakshmi, 2021). Using deep learning models to frame the issue as a supervised learning problem, GANs are a revolutionary method for training generative models. Automatic pattern recognition is a capability of GANs. The discriminator and the generator are two sub-models of a GAN. In contrast to the discriminator model, which is used to identify whether generated samples are authentic or fraudulent to produce new instances, the generator model is trained (Mahalakshmi et al., 2024). The primary contributions of this paper are:

- A deep learning framework for sentiment analysis on Twitter called the Conditional GAN (CGAN) is suggested in this research.
- With a deep learning approach, CNN extracts features from Twitter data.
- Long Short-Term Memory (LSTM) has been incorporated into the discriminator to enhance the classification model's accuracy.

The work is organized as follows: The related works are covered in Section 2. Section 3 presents the proposed CGAN. Section 4 presents the results of the study and discussion. Section 5 offers the conclusion in the final part.

2. Literature survey

Today's society is pervasive, and everyone uses social media daily. With social media data, many studies and statistics can be performed. Using either a lexicon-based strategy using WordNet and its POS (parts of speech) or a lexicon-based strategy using SentiWordNet, an algorithm has been made to pick emotional words from a phrase that makes sense in this situation and is given a sentiment polarity (Gopalan et al., 2023). In Gupta et al. (2021), a prediction model was developed to forecast a person's awareness of safety precautions in five significant Saudi Arabian regions. In Venkataramanan et al. (2021), the authors proposed a multilingual Twitter sentiment analysis (MLTSA). The MLTSA method was used in this work to solve these two problems. The MLTSA algorithm was divided into two sections. One example is utilizing NLP to recognize and translate non-English tweets into English. Second, a suitable pre-processing approach with NLP assistance can minimize data sparsity.

The authors introduced a unique feature reduction method based on a genetic algorithm (GA) (Hassonah et al., 2020). Without sacrificing accuracy, the hybrid technique can reduce the feature set size by up to 42 %. The authors also recommended a particular cross-disciplinary geopolitical location for our sentiment research methodology as a case study. In Iqbal et al. (2019), the authors investigated several sentiment measurement feature sets and classifiers. On top of that, a feature set is used to compare the real-world performance of modern deep learning algorithms, ensemble-based techniques, and standard machine learning methods. According to the estimated results, different feature sets affect classifier performance in sentiment quantification.

This study aimed to ascertain the public's perception of the statewide lockdown imposed by the Indian government in an attempt to contain the coronavirus's spread (Jin et al., 2020). This study analyzed the sentiment of tweets written by Indians using natural language processing (NLP) and machine learning classifiers. Following the emergence of new COVID-19 cases in India, the LSTM-based recurrent neural network (RNN) was proposed as a sentiment analysis technique (Neelakandan & Paulraj, 2020). The system can represent multiple emotions simultaneously because it uses multi-label sentiment categorization. Pre-trained word embedding has been used in the Bidirectional-LSTM (Bi-LSTM) method to reduce the size of the text representation and avoid data sparsity issues (Manikandan et al., 2023). An attention mechanism has also been implemented, which involves giving words and sentence weights to gather n-gram characteristics and focus on the most critical context-relevant information.

Combining numerous weak learner base classifiers into one ensemble classification is a novel ensemble classifier approach (Patel & Passi, 2020). The recommended ensemble technique surpassed all individual classifiers on the most widely used sentiment analysis datasets, significantly improving sentiment classification performance. A method for identifying tweets as extremist or non-extremist was developed to analyze terrorism-related content (Patra et al., 2015). By Phan et al. (2020), a lexicon-enhanced LSTM model was introduced. Before training a word sentiment classifier, the model uses the sentiment lexicon as supplementary input. Then, it retrieves the sentiment embedding of words not in the lexicon (Rosa et al., 2019). It contained an extensive context that more accurately portrayed the material on paper and a dependency structure that increased the precision of token-token semantic learning.

The RNN was used to develop appropriate solutions for various issue sizes, considering the problem's NP-hardness. They used the Windowed Multivariate Autoregressive Model (WMAR) and Independent Component Analysis (ICA) to preprocess the data to identify possible features. Self-adaptive probabilistic neural networks are offered as an alternative to probabilistic neural networks with self-learning techniques (Chandramohan et al., 2023). Self-adaptive probabilistic neural networks were tested and trained using the best-selected spread. The suggested self-adaptive probabilistic neural network was further enhanced by incorporating two condensed techniques. In Tam et al. (2021), the authors were the first to demonstrate an ELM with a self-adapting mechanism. It was suggested in this work that the self-adaptive extreme learning machine (SaELM), a unique variation of the ELM, be used. The optimal number of neurons to create neural networks is chosen using a self-adaptive learning algorithm called SaELM in the hidden layer. Parameter adjustments are not necessary throughout the training procedure.

By proposing a new measure of user similarity that considers cognitive aspects like decision-making style and preference consistency, (Wang et al., 2016) offers a unique strategy for enhancing the performance of collaborative filtering-based recommendation systems. They developed the Efficient Gowers-Jaccard-Sigmoid Measure (EGJSM), a similarity metric that penalizes unfavorable evaluations by combining the Jaccard and Gower coefficients with a nonlinear sigmoid function. On benchmark datasets, experiments revealed that the suggested strategy performed better than several already-used techniques (Reka et al., 2023). A unique deep-learning technique has been proposed to enhance the identification of malware variants (Nilabar Nisha et al., 2023). Deep learning performed exceptionally well in picture recognition in earlier studies. The harmful code was transformed into grayscale photos using the suggested detection technique (Yi et al., 2016). The CNN was then used to identify and categorize the photographs, utilizing the ability to automatically extract the characteristics of the malware images (Zuo et al., 2020).

In this study, the Barnacles Mating Optimizer (BMO) optimizes these parameters automatically (Mustaffa & Sulaiman, 2023). As a relatively new optimization algorithm, it effectively addresses various optimization problems. Multi-class Sentiment Analysis (SA) is an essential field of computational linguistics that extracts multiple opinions expressed in a text using NLP and text-mining techniques (Haque et al., 2023). Fake news has been a concern worldwide, and social media has only amplified this phenomenon. Fake news has been affecting the world on a large scale, as these are targeted to sway the crowd's decisions in a particular direction (Palaniappan & Annamalai, 2019). Following the content-based classification approach, this paper proposes a model for fake news classification based on news titles. The model uses a BERT model with its outputs connected to an LSTM layer. This paper aims to determine the sentiments expressed via texts on social media using machine learning methods (Demircan et al., 2021).

References	Datasets	Algorithms	Performance	Limitation
Ahmad et al. (2019)	Consuming streaming data	CNN, LSTM, FastText, and GRU	Accuracy is 92.66 %	The data collected for the study is biased, leading to inaccurate or incomplete conclusions.
Aljameel et al. (2020)	Twitter dataset	SVM, KNN, and Naïve Bayes (NB)	accuracy of SVM is 85 %, KNN is 64 % and NB is 80 %	The study only examined the association between awareness of precautionary procedures and sentiment toward COVID-19. It did not establish a causal relationship between the two variables.
Arun and Srinagesh (2020)	Multilingual Twitter data	MLTSA and SVM	Accuracy of 95 %.	The paper mainly focused on using pre-trained word embeddings as input features for the model. While this approach can work well for some tasks, more is needed for capturing the nuances of sentiment in different languages. Advanced feature engineering methods have enhanced the model's performance.
Ayyub et al. 2020	Stanford Twitter, TS-Gold, Sanders	Naïve Bayes, Decision Tree, SVM	Accuracy is 63 %	The study used a small dataset review, which only represents some reviews. It could limit the generalizability of the findings to other datasets or domains.
Wang et al. (2020)	MR data set, TREC data set,AG News data set, Twitter data set, SST-2 data set	CNN	87.37 % of Accuracy	It is, therefore, challenging to assess the sensitivity of the suggested technique to these hyperparameters.
Chandra et al. (2021)	Senwave COVID-19 sentiment data	LSTM	–	The specific events that have influenced their sentiment toward COVID-19.
Bhuvaneshwari et al. (2022)	Amazon customer review dataset	Bi_LSTM	Accuracy of 89 %	The study neglects the topic of the proposed model's sensitivity to hyperparameters, such as the number of filters, learning rate, and batch size. It is important information for practitioners who want to replicate the model.
Hama Aziz and Dimililer (2021)	SemEval-2017	SentiXGboost	Accuracy of 90.8 %	The paper focuses on the sentiment analysis of social media posts, which has a relatively narrow scope. The findings may not be generalizable to other types of text data or other applications of sentiment analysis.
Neelakandan and Paulraj (2020)	Twitter dataset	Gradient-Boosted Decision Tree	F1 score of 92.04 %	The authors used a limited set of features, which only capture some of the nuances of sentiment in Twitter data. For example, they did not consider the use of emojis, which are commonly used to express sentiment on Twitter.

3. Methods

This work analyses the sentiments of tweets as positive or negative. The CGAN network is used for sentiment analysis to classify the tweets based on the sentiment score. The following are the several stages of the framework used for sentiment analysis. The sentiment analysis's architecture is depicted in Fig. 1.

The above figures show the flow of the proposed work, in which the dataset is preprocessed using tokenization, word removal, hashtag and username removal, punctuation removal, stemming, and lemmatization. After preprocessing the data, the feature extraction procedure extracts the features from the dataset. After that, using the preprocessed data, the CGAN model is trained to categorize tweets as positive or negative.

3.1. Dataset

In this work, the dataset used is US Election 2020 Tweets, which is used to analyze the tweets based on the election. This dataset contains many features like tweet_id, tweet, retweet_count, etc. Table 1 displays the characteristics of the dataset.

3.2. Data preprocessing

Many redundant and unnecessary details can be found in the input tweets. Unwanted symbols were removed from the tweets using preprocessing methods to sanitize them. This step's main objective was to get the data ready for incorporation into the models made for training and validation. A few of the preprocessing techniques used in this work are listed below.

3.2.1. Tokenization

By using this method, a lengthy text file was divided into manageable, smaller portions. Tweets are particularly captivating since hashtags, emoticons, and other fascinating symbols have varying meanings. The building block for a phrase or paragraph is the single entity token. The token is a random data string with no meaningful or exploitable value, and it can be obtained as follows:

['@meiselasb',' I', 'wonder',' which', 'drugs','#Trump',' takes?', 'That is',' not',' only', 'masses',' of', 'burgers', 'Maybe',' he is ', '.', '.' '.']

3.2.2. Stop words removal

Stop word removal is one of the preprocessing techniques that are most often employed throughout NLP applications. Words appearing in all the corpus's papers are to be removed. Stop words, including conjunctions, prepositions, articles, and pronouns that do not affect the phrase's meaning. By eliminating these phrases from the text, special attention was given to the critical information while removing the unnecessary information.

3.2.3. Hashtags and username removal

A hashtag is a type of metadata tag that enables users to construct dynamic, user-generated tags for tweets about a particular subject on several social media platforms. The raw input tweet contains usernames with the prefix "@" and hashtags with the symbol "#" as a prefix. Most of the time, the hashtags and usernames are optional for analysis.

['I', 'wonder',' which', 'drugs', 'takes?', 'That is',' not',' only', 'masses',' of', 'burgers', 'Maybe',' he is ', '.', '.' '.']

3.2.4. Punctuation removal

Text processing techniques such as removing inconsistencies are the second most common. By eliminating punctuation, we will be able to treat each text equally. For example, the words data and data! After punctuation is removed, they are treated equally. Keeping multiple punctuation marks in a word does not contain vital information, so we should remove the repetition. The tweet data has various punctuation, which should be released because it is unnecessary to analyze the tweets.

3.2.5. Stemming

Stemming converts a word to a stem that becomes its suffixes,

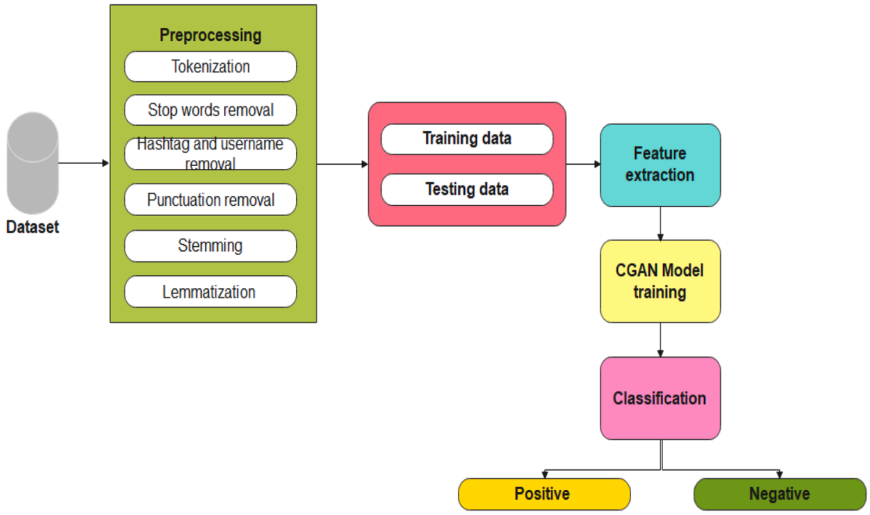

Fig. 1. Sentiment analysis architecture.

Table 1
Features of the dataset (Divyapushpalakshmi & Ramalakshmi, 2021).

Name of the Feature	Description
Likes	Number of Likes
User_screen_name	Screen name of tweet creator
tweet	Full tweet text
Source	Utility used to post tweet
Long	Longitude parsed from user_location
User description	Description of self by tweet creator
User_name	Username of tweet creator
country	Country parsed from user_location
State_code	State code parsed from user_location
User_join_date	Join date of tweet creator
User_followers_count	Followers count on tweet creator
Created-at	Date and time of tweet creation
User_id	User ID of tweet creator
tweet_id	Unique ID of the tweet
Collected at	Date and time tweeet data was mined from twitter
city	City parsed from user_location
state	State parsed from user_location
Lat	Latitude parsed from user_location
User location	Location given on tweet creator's profile
retweet_count	Number of retweets

prefixes, or roots. NLP and natural language understanding both depend on originating. The procedure of eliminating affixes (circumfixes, suffixes, prefixes, and infixes) from a word to get the word stem. Words are stemmed to change different word forms into the same accepted canonical form. For instance, "sings" and "singing" are combined to get the word sing. This technique enhances the feature-extraction process. The benefits of stemming are its ease of use and speed.

3.2.6. Lemmatization

A word's meaning-filled basic form, called a lemma, is changed into context awareness when it is lemmatized. Although the goal of stemming a word is the same, the term's true meaning is often hidden. Lemmatization, which involves using vocabulary and looking at the morphology of words, is frequently used to describe doing tasks correctly. Lemmatization's key benefit is that it considers the word's context to identify the intended meaning the user is seeking. Thanks to this technique, the user's work was completed faster and with less noise.

3.3. Sentiment score

The number of positive and negative phrases used in each tweet was counted to get the sentiment score. The sentiment score identifies emotions and assigns them sentiment ratings. Each positive phrase counts as +1, and each negative word counts as −1 to compute the emotion score. Each tweet is posted with a sentiment score. The semantic orientation ranges are positive and negative, depending on the emotion score, as indicated in Table 2.

3.4. Feature extraction using CNN

The hardware implementation of convolutions on a graphical processor unit (GPU) is a crucial part of computer graphics. Applications like text classification and sentiment analysis do not need the data's sequential nature to preserve the information. For feature extraction and classification, it is helpful to use CNN, a neural network that can send data to the following layer without losing spatial information. The CNN model could be good enough in terms of computation. Convolution is a crucial component of the CNN model, successfully finding tweets' most essential terms or phrases.

The convolutional layer receives input from the word matrix, representing each token in each row. The input matrix is subjected to a series of convolutional filter applications to produce a variety of feature maps illustrating significant patterns of input data. With this change, the dimensionality is decreased, yet the essential qualities are still captured. The semantic similarity of words in a phrase is one of these CNN properties that is suited for using geographic information. The convolutional and pooling layers gather the input data attributes and then transfer them to the feature map. The convolutional operation on the matrix must be utilized to produce new features from the input text.

A bias and a ReLU activation function are combined to produce new features. Several filters with tunable parameters make up the convolutional layer. These filters can acquire greater activation levels for features in this layer by varying their weight values. The filter's height is h, and its width is fixed at the number a, corresponding to the word vector's dimension. A filter v, constructed in Eq. (1), extracts features like si.

$$s_i = f\left(v.\left[w_i : w_{i+h-1}\right] + b\right) \quad (1)$$

In which f indicates a nonlinear function, The activation function is ReLU, and the(.) operator indicates the convolution operation, b represents a bias term, a sequence of words of length hi, and wi represents a word.

A feature set F was produced by sliding the filter window, which is described in Eq. (2) as follows, using filters of various heights in the convolution operation:

$$F = \left[s_1, s_2, s_3 \ldots s_n\right] \quad (2)$$

Here,n is the size of the set F, and *xi* is the feature vector produced by each filter's convolution operation. To generate the pooled feature vector x_W, as described in Eq. (3), each feature vector xi is a weighted sum according to its weight W_i.

$$s_W = \sum_{1}^{n} W_i s_i \tag{3}$$

3.5. Classification using CGAN

A GAN is an innovative approach that combines generational and discriminant models. The discriminant model assesses if the results of the created Twitter data are consistent with the original distribution, while the generation model learns the distribution of the original Twitter data. After modifying its learning settings, the actual distribution remains the most appropriate. The emergence of the GAN enables the development of a novel unsupervised learning approach for feature extraction. The samples are continually trained against each other in the overall model, and the generator parameters are updated based on the discriminator's gradient feedback, which is independent of the distribution of the input data samples. GANs have made significant strides in computer vision, making them the most sought-after generative model. Fig. 2 depicts the fundamental structure of GAN.

3.6. The generator

The GAN network comprises the discriminator (D) and generator (G) models, which are two separate models. Using the latent variable z, G creates fake Twitter samples similar to the actual Twitter data space. D chooses whether G or the virtual data space is used as its input. "Adversarial" sums up G and D's pursuit of their objectives. D wants to categorize samples as real or fake. L (G, D) is therefore regarded as an objective function that is a part of the classification issue. According to D, the output will be optimized if a sample is drawn from accurate data. Conversely, if a sample comes from G, D will produce less.

3.7. The discriminator

The discriminator network is utilized to ascertain whether the produced Twitter samples are legitimate. Usually, 50 % of the output sample must be classified as false for the discriminator to perform optimally. The training of the discriminator to maximize is the initial stage of constructing GAN networks. The loss function of GANs is represented mathematically in the following, as indicated in Eq. (4).

$$\underset{G}{min}\ \underset{D}{max}\ L(G,D) = \underset{G}{min}\ \underset{D}{max}\ E_{s\sim q_{data}}[\log D(s) + E_{z\sim q_z}[\log(1-D(G(z)))] \tag{4}$$

Where D(s) is the estimator of the discriminator's probability that the actual data instance s is confirmed, the expected value for all genuine data instances is Es. When fed noise z, the generator produces G(z). The symbol D(G(z)) estimates the discriminator's likelihood that a false instance is actual. Here, Ez is the expected value over all random inputs to the generator (effectively, the expected value over all created fictitious instances G(z)), where s represents a sample from a distribution in latent space. Qz (z), where z is drawn from the actual distribution of the dataset data (s).

The generator will use the conditional argument to produce synthetic samples. Twitter sentiment score is the requirement. The tweet is seen as favorable if the score is 1. The tweet is regarded as unfavorable if the score is −1. The strategies used by CGAN and GAN are similar. The discriminator and the generator have preconditioned an extra input (y). This conditioning was accomplished by including a second input layer and feeding it into the generator and discriminator. "Y" stands for any extra information. The conditional GAN can generate fake samples corresponding to the category indicated by the dependent input used to classify the jobs.

Table 2
Sentiment score with semantic orientation.

Semantic orientation	Sentiment score
Positive	1
Negative	−1

$$\underset{G}{min}\ \underset{D}{max}\ L(G,D) = \underset{G}{min}\ \underset{D}{max}\ E_{s\sim q_{data}}[\log D(s|r) + E_{z\sim q_z}[\log(1-D(G(z|r)))] \tag{5}$$

Like the standard GAN loss function, the above-modified loss function engages in a min-max game. The idea behind conditional information, y, is that by including more data, such as emotion score, Both the discriminator D and the generator G would pick up on specific patterns and operate in them, as illustrated in Fig. 3.

3.8. LSTM in discriminator

An RNN variant known as an LSTM learns short- and long-term correlations from sequential data. An LSTM network often comprises many repeating modules or LSTM units to maximize network performance. When information is transferred from one LSTM unit to another, they concatenate. Three different sorts of gates are included in each unit: forget, which selects which categories of old data should be deleted; update, which specifies which new data should be added; and output, which determines which types of new data should be produced. The LSTM network's use in the discriminator is shown in Fig. 4.

C_{t-1} transmits the information to C_t, whose inner operands will alter. Input $_t$, output y_t, and latent variable Ct are all used. Assuming W and b are parameters that need to be estimated to minimize some loss functions, the mathematical processes in the three gates are stated as follows:

Forget gate:

$$F_t = sigmoid\left(W_f[y_{t-1}, X_t] + b_f\right) \tag{6}$$

Update gate:

$$\begin{aligned} U_t &= sigmoid(W_u[y_{t-1}, X_t] + b_u), \\ \widetilde{C}_t &= tanh(W_c[y_{t-1}, X_t] + b_c), \\ C_t &= F_t * C_{t-1} + U_t * \widetilde{C}_t \end{aligned} \tag{7}$$

Output gate:

$$\begin{aligned} O_t &= sigmoid(W_o[y_{t-1}, X_t] + b_o) \\ y_t &= O_t * \tanh(C_t) \end{aligned} \tag{8}$$

The sigmoid and tanh are elementwise activation functions applied to an input vector, whereas * denotes vector multiplication. These three gates that work together determine the ultimate information output from a single unit. Memory characteristics have been utilized in this study to increase the accuracy of the LSTM.

4. Results and discussion

This section outlines the sentiment analysis findings using the suggested CGAN model. The proposed model and the currently used algorithms are contrasted in the sentiment analysis of Twitter data. About 20 % of the dataset was validated using the built-in feature extraction-classifier model combinations. The other 80 % was used for training. Evaluation metrics, including the F1 score, precision, accuracy, and recall, evaluate how well the suggested model performs.

Accuracy (A): It is the number of correctly classified tweets out of the total number of tweets.

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

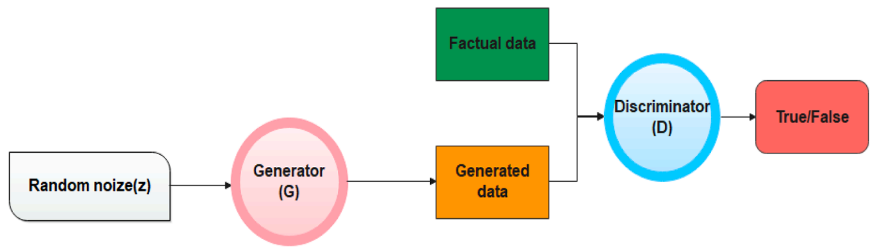

Fig. 2. Basic structure of GAN.

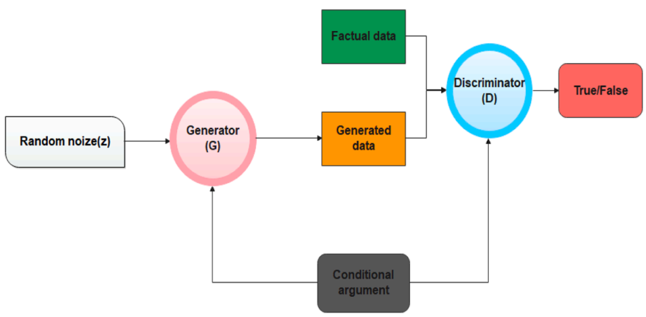

Fig. 3. Structure of conditional GAN.

Precision (P): It measures the exactness of the classification classifier. The number of incorrect classifications is known as precision.

$$P = \frac{TP}{TP + FP}$$

Recall (R): It is the measure of completeness or sensitivity of the classifier. The recall is the number of correct classifications penalized by the number of incorrect classifications.

$$R = \frac{TP}{TP + FN}$$

F1 score: One way to express the F-measure is as a weighted harmonic mean of recall and precision.

$$F1 = \frac{2(PXR)}{P + R}$$

From Table 3 and Fig. 5, the experiment results demonstrate that the new CGAN algorithm performs better in accuracy than the current methods of CNN, LSTM, and Bi-LSTM. The accuracy of the CGAN is 93.33 %, which is a lot higher than the accuracy of the other methods.

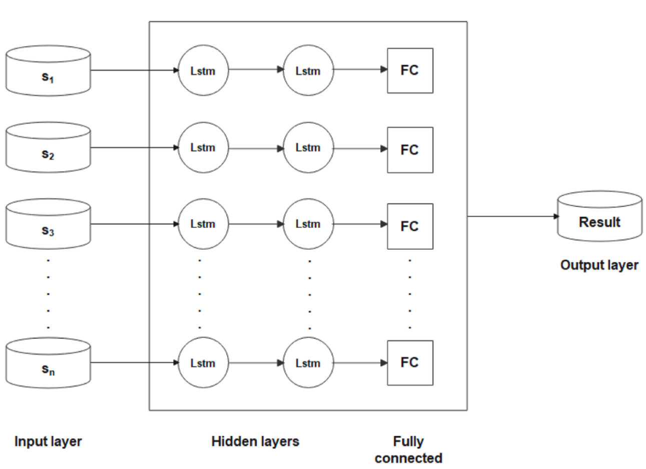

Fig. 4. Structure of discriminator.

However, one drawback of the current algorithms is that they need to be able to capture the relationships between the input information. For instance, in the input data, CNN concentrates on local patterns but cannot detect long-term relationships. The LSTM and Bi-LSTM can capture longer-term dependencies, although these algorithms need to model complicated interactions between the input characteristics better. The proposed CGAN approach uses a GAN to learn the underlying data distribution and produce artificial samples to overcome this constraint.

The CGAN increases the classification task's accuracy by better capturing the correlations between the input characteristics by including the synthetic samples in the training process. In addition, using a GAN makes it possible to create new synthetic samples comparable to the original data, which is useful when the actual data is scarce or hard to get.

It is evident from Table 4 and Fig. 6 that the suggested approach has the highest precision when compared to the current methods. The findings demonstrate that the proposed algorithm, CGAN, beats the existing methods with an accuracy rate of 92.86 %, showing that CGAN is a viable approach for this work. When comparing the current algorithms, CNN had the lowest precision rate of 85.98 %. That is because CNNs are more suitable for tasks that involve spatial information, such as image recognition, than sequential data like text. LSTM and Bi-LSTM had higher precision rates of 91.45 % and 91.55 %, respectively, indicating that they are better suited for sequential data. Still, it is hard for LSTMs to learn long-term relationships in the data because of the vanishing gradient problem. Bi-LSTMs were developed to solve this issue by processing data forward and backward. However, they need more computational resources.

Table 5 and Fig. 7 illustrates how the suggested algorithm, CGAN, has outperformed the existing algorithms in Recall, achieving an excellent score of 93.11 %. Regarding existing algorithms, CNN, LSTM, and Bi-LSTM have relatively lower recall scores, ranging from 87.63 % to 90.29 %. These scores suggest that these models only effectively identify some relevant data points, potentially leading to missed or misclassified information as in Table 6. The proposed algorithm, CGAN, has achieved a recall score of 93.11 %, indicating that it is more effective at identifying relevant data points than the existing algorithms. This advantage could be precious when comprehensive and accurate data analysis is essential. Additionally, the use of a CGAN allowed for more nuanced analysis, as it can generate synthetic data that can be used to fill gaps or provide additional context.

Fig. 7 indicates that the CGAN algorithm, which scored 92.68 % on the F1 scale, outperformed CNN, LSTM, and Bi-LSTM. LSTM came in second with an 85.47 % score, ahead of CNN with 86.6 %. It indicates

Table 3
Accuracy comparison of proposed work with the existing work.

Algorithms	Accuracy (%)
CNN	89.56
LSTM	90.87
Bi-LSTM	91.78
CGAN	93.33

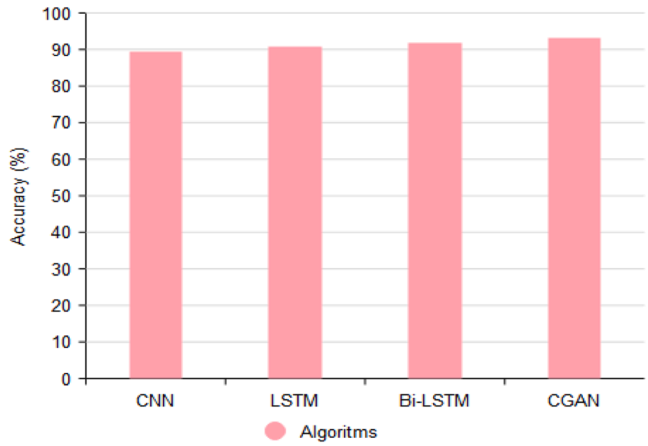

Fig. 5. Comparison of the accuracy of the proposed algorithm with the existing algorithm.

Table 4
Comparison of precision between proposed and existing algorithms.

Algorithms	Precision (%)
CNN	85.98
LSTM	91.45
Bi-LSTM	91.55
CGAN	92.86

that the CGAN algorithm outperformed the other three algorithms in classifying Twitter data. The high F1 score that CGAN obtained suggests that it could accurately organize the Twitter data and produce realistic new data. The time taken with various tweets is compared in Table 7 for performance. Compared to other classifiers with the exact tweet count, the total time required by multiple classifiers for that number of tweets is adequate in CGAN.

From Table 7 and Fig. 8 illustrates an analysis of CNN, LSTM, and Bi-LSTM about the number of tweets and the required time. Fig. 8 shows an analysis of the GAN classifier for the number of tweets and time taken. The findings represent the time, measured in seconds, that various classifiers needed to analyze Twitter data. The time required for each of the classifiers used for analysis—CNN, LSTM, Bi-LSTM, and CGAN—is provided for various sample sizes of 500, 1000, 1500, 2000, and 2500.

The classifier known as CNN requires the longest processing time for analyzing Twitter data. For a sample size of 500, it takes 0.056 s; for a sample size of 1000, 0.072 s; for a sample size of 1500, 0.12 s; for a sample size of 2000, 0.138 s; and for a sample size of 2500, it takes 0.189 s. As the sample size increases, CNN's processing time increases as well. LSTM requires less time to analyze Twitter data than CNN. A sample size of 500 requires 0.038 s, a sample size of 1000 requires 0.059 s, a sample size of 1500 requires 0.083 s, a sample size of 2000 requires 0.125 s, and a sample size of 2500 requires 0.173 s. With a larger sample size, LSTM takes longer, but it still takes less time than CNN.

For analyzing Twitter data, Bi-LSTM, also known as Bidirectional Long Short-Term Memory, takes almost as long as LSTM. It takes 0.042 s for 500 samples, 0.61 s for 1000 samples, and for a sample size of 1500, 0.081 s, 0.128 s for 2000 samples, and 0.168 s for 2500 samples. Although the time required by Bi-LSTM increases along with the amount of the sample, it is still like LSTM.

The CGAN is a classifier that quickly analyses Twitter data. A sample size of 500 requires 0.035 s, a sample size of 1000 requires 0.058 s, a sample size of 1500 requires 0.069 s, a sample size of 2000 requires 0.117 s, and a sample size of 2500 requires 0.153 s. While CGAN's processing time increases with sample size, it is still much slower than that of CNN, LSTM, and Bi-LSTM. According to the examination of

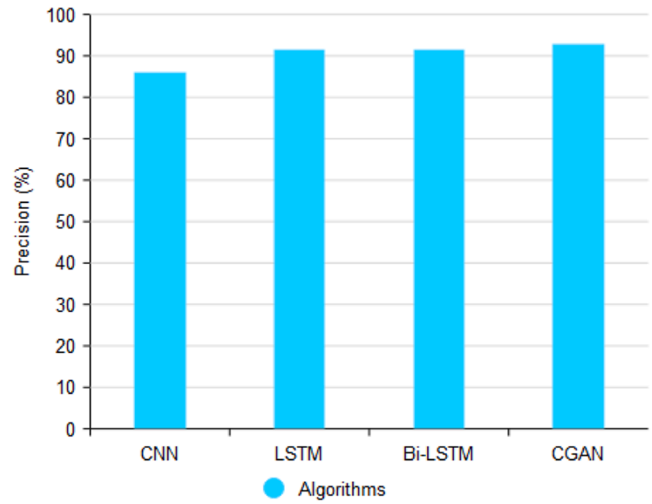

Fig. 6. shows the precision comparison between the proposed and existing algorithms.

Table 5
Illustrates the recall comparison between the proposed algorithm and the existing algorithm.

Algorithms	Recall (%)
CNN	87.63
LSTM	89.5
Bi-LSTM	90.29
CGAN	93.11

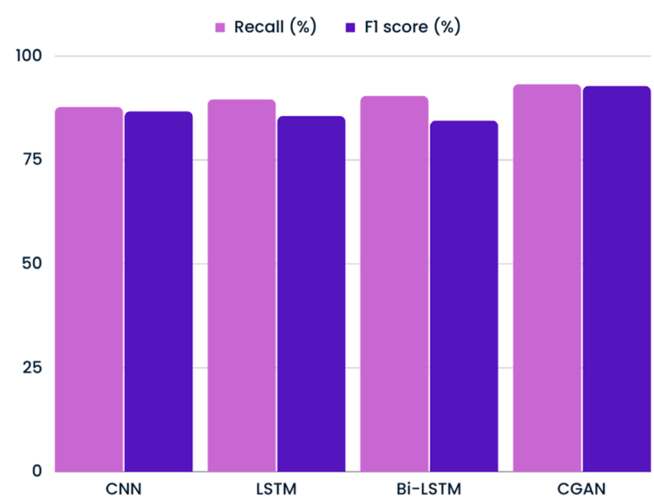

Fig. 7. Recall & F1-Score Comparison between the Proposed and Existing Algorithms.

Twitter data, CGAN is the fastest classifier, whereas CNN requires the most significant processing time. LSTM and Bi-LSTM run virtually simultaneously, taking longer than CGAN but less than CNN. According to these findings, CNN is the least effective classifier regarding the time required, whereas CGAN, LSTM, and Bi-LSTM are the most effective classifiers for analyzing Twitter data.

Table 8 lists the LSTM's top hyperparameters, and conducting hyperparameter adjustments is essential to achieving good model performance. Therefore, the hyper-parameter was tuned, and the accuracy was optimized using the randomized search approach. The sentiment analysis hyperparameters suggested for the proposed model are the learning rate, hidden layers, activation function, dropout rate, batch

Table 6
Comparison of F1 score between proposed and existing algorithms.

Algorithms	F1 score (%)
CNN	86.6
LSTM	85.47
Bi-LSTM	84.36
CGAN	92.68

size, and epochs.

Specific hyperparameters are crucial for the neural network-based sentiment analysis models to operate at their best. The learning rate, a hyperparameter, controls the step size during each iteration of the optimization process. The model can converge fast and stay out of local minima by using the suggested learning rate of 0.01, which is a modest number. Hidden layers in the neural network represent various layers, and the recommended value of 2 is appropriate for sentiment analysis.

Although overfitting is also more likely, adding more hidden layers improved the model's capacity to learn complex patterns. The Rectified Linear Unit (ReLU) is a well-liked option for sentiment analysis, and the activation function is employed to introduce non-linearity in the neural network. This straightforward function, which sets all negative numbers to zero, has been shown to perform admirably in various NLP applications. A part of the neurons was removed during training randomly by the regularization strategy called dropout rate to eliminate the problem of overfitting.

An acceptable amount for sentiment analysis is the recommended value of 0.5, which denotes that half of the neurons will be eliminated during training. The batch size is the number of samples utilized for each training iteration, and the suggested value of 128 provides a reasonable compromise between training speed and model quality. The epoch quantity indicates how often the neural network feeds the training dataset.

The suggested value of 30 is an excellent quantity to provide enough training without overfitting the model. The included hyperparameters for the proposed sentiment analysis are:

- Two hidden layers.
- A moderate learning rate.
- A 0.5 dropout rate.
- A ReLU activation function.
- 30 epochs.
- A batch size of 128.

With these hyper parameters' help, the model learns complicated patterns while maintaining an appropriate balance between avoiding overfitting and learning complex patterns.

5. Conclusions

Twitter is the most widely used social media platform for information sharing. This paper proposes a CGAN for sentiment analysis on Twitter. The work's main contribution is the use of CNN in the GAN generator and LSTM in the GAN discriminator. A few key hyperparameters must be set for the neural network-based sentiment analysis models to perform as well as possible. The learning rate is one hyperparameter that regulates the step size in each optimization iteration. The included hyperparameters for the proposed sentiment analysis are:

- Two hidden layers.
- A moderate learning rate.
- A 0.5 dropout rate.
- A ReLU activation function.
- 30 epochs.
- A batch size of 128.

With these hyperparameters' help, the model learns complicated patterns while maintaining an appropriate balance between avoiding overfitting and learning complex patterns. The accuracy of the CGAN is 93.33 %, which is a lot higher than the accuracy of the other methods. According to the examination of Twitter data, CGAN is the fastest classifier, whereas CNN requires the most significant processing time. LSTM and Bi-LSTM run virtually simultaneously, taking longer than CGAN but less than CNN. The CGAN is the classifier that quickly analyses Twitter data. A sample size of 500 requires 0.035 s, a sample size of 1000 requires 0.058 s, a sample size of 1500 requires 0.069 s, a sample size of 2000 requires 0.117 s, and a sample size of 2500 requires 0.153 s. The CGAN increases the classification task's accuracy by better capturing the correlations between the input characteristics by including the synthetic samples in the training process.

Table 7
Compares the time consumption of tweets between the proposed and existing classifiers.

Classifiers	Time in *sec* (s)				
	500	1000	1500	2000	2500
CNN	0.056	0.072	0.12	0.138	0.189
LSTM	0.038	0.059	0.083	0.125	0.173
Bi-LSTM	0.042	0.061	0.081	0.128	0.168
CGAN	0.035	0.058	0.069	0.117	0.153

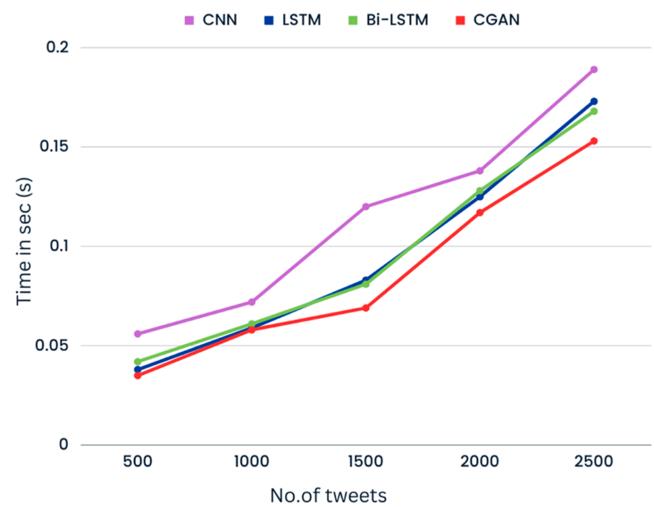

Fig. 8. Shows the Time consumption for several tweets.

Table 8
The best hyperparameters and the values of LSTM.

Hyperparameters	Values
Learning rate	0.01
Hidden layers	2
Activation function	ReLU
Dropout	0.5
Batch size	128
Epochs	30

List of abbreviations

CNN	Convolutional Neural Network
SWN	Sent WordNet
CGAN	Conditional Generative Adversarial Network
I-EHO	Improved Elephant Herd Optimization
TSA	Tweet Sentiment Analysis

Declarations

Availability of information and resources

Data sharing does not apply to this article since no datasets were created or analysed for the present investigation.

Funding

No government or private concern received funding.

Competing interests

The authors declare that they have no competing interests.

CRediT authorship contribution statement

V. Mahalakshmi: Visualization. **P. Shenbagavalli:** Visualization. **S. Raguvaran:** Visualization. **V. Rajakumareswaran:** Formal analysis. **E. Sivaraman:** Methodology.

Declaration of competing interest

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

Acknowledgements

Not applicable.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ijcce.2024.03.002.

References

Abdulsaheb, G., Selvaraj, S., Manikandan, A, Palanisamy, S., Uddin, M., Khalaf, I., … Khalaf, O. (2023). Reverse polarity optical Orthogonal frequency Division Multiplexing for High-Speed visible light communications system. *Egyptian Informatics Journal, 24*, 100407. https://doi.org/10.1016/j.eij.2023.100407

Ahmad, S., Asghar, M. Z., Alotaibi, F. M., & Awan, I. (2019). Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Human-Centric Computing and Information Sciences, 9*(1).

Al-Zoubi, A. M., Alqatawna, J., Faris, H., & Hassonah, M. A. (2021). Spam profile detection on social networks using computational intelligence methods: The effect of the lingual context. *Journal of Information Science, 47*(1), Article 01655515.

Aljameel, S. S., Alabbad, D. A., Alzahrani, N. A., Alqarni, S. M., Alamoudi, F. A., Babili, L. M., et al. (2020). A sentiment analysis approach to predict an individual's awareness of the precautionary procedures to prevent COVID-19 outbreaks in Saudi Arabia. *International Journal of Environmental Research and Public Health, 18*(1), 218.

Ayyub, K., Iqbal, S., Munir, E. U., Nisar, M. W., & Abbasi, M. (2020). Exploring diverse features for sentiment quantification using machine learning algorithms. *IEEE Access : Practical Innovations, Open Solutions, 8*, 142819–142831.

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Generation Computer Systems, 115*, 279–294.

Bommaraju, K., Manikandan, A., & Ramalingam, S. (2017). Aided system for visually impaired people in bus transport using intel galileo gen-2: technical note. *International Journal of Vehicle Structures and Systems, 9*(2), 110–112. https://doi.org/10.4273/ijvss.9.2.09

Ce, P., & Tie, B. (2020). An analysis method for the interpretability of CNN text classification models. *Future Internet, 12*(12), 228.

Chandra, R., & Krishna, A. (2021). COVID-19 sentiment analysis via deep learning during the rise of novel cases. *PloS one, 16*(8), Article e0255615.

Chandramohan, K., Manikandan, A., Ramalingam, S., & Dhanapal, R. (2023). Performance evaluation of VANET using directional location aided routing (D-LAR) protocol with sleep scheduling algorithm. *Ain Shams Engineering Journal.* , Article 102458. https://doi.org/10.1016/j.asej.2023.102458

Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G. G., & Chen, J. (2018). Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics, 14* (7), 3187–3196.

Demircan, M., Seller, A., Abut, F., & Akay, M. (2021). Developing Turkish sentiment analysis models using machine learning and E-commerce data. *International Journal of Cognitive Computing in Engineering, 2*. https://doi.org/10.1016/j.ijcce.2021.11.003

Divyapushpalakshmi, M., & Ramalakshmi, R. (2021). An efficient sentimental analysis using hybrid deep learning and optimization techniques for Twitter using parts of speech (POS) tagging. *International Journal of Speech Technology, 24*(2), 329–339.

Gopalan, S. H., Ashok, J., Manikandan, A., et al. (2023). Data dissemination protocol for VANETs to optimize the routing path using hybrid particle swarm optimization with sequential variable neighbourhood search. *Telecommunication Systems*. https://doi.org/10.1007/s11235-023-01040-2

Gupta, P., Kumar, S., Suman, R. R., & Kumar, V. (2021). Sentiment analysis of lockdown in India during COVID-19: A case study on Twitter. *IEEE Transactions on Computational Social Systems, 8*(4), 992–1002.

Haque, R., Islam, N., Tasneem, M., & Das, A. (2023). Multi-class sentiment classification on Bengali social media comments using machine learning. *International Journal of Cognitive Computing in Engineering, 4*. https://doi.org/10.1016/j.ijcce.2023.01.001

Hassonah, M. A., Al-Sayyed, R., Rodan, A., Al-Zoubi, A. M., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowledge-Based Systems, 192*, Article 105353.

Iqbal, F., Hashmi, J. M., Fung, B. C., Batool, R., Khattak, A. M., Aleem, S., et al. (2019). A hybrid framework for sentiment analysis using genetic algorithm-based feature reduction. *IEEE Access : Practical Innovations, Open Solutions, 7*, 14637–14652.

Jin, N., Wu, J., Ma, X., Yan, K., & Mo, Y. (2020). Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification. *IEEE Access : Practical Innovations, Open Solutions, 8*, 77060–77072.

Mahalakshmi, G., Ramalingam, S., & Manikandan, A. (2024). An energy efficient data fault prediction based clustering and routing protocol using hybrid ASSO with MERNN in wireless sensor network. *Telecommunication Systems*. https://doi.org/10.1007/s11235-024-01109-6

Manikandan, A., Madhu, G. C., Flora, G. D., et al. (2023). Hybrid Advisory Weight based dynamic scheduling framework to ensure effective communication using acknowledgement during Encounter strategy in Ad-hoc network. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-023-01421-5

Mustaffa, Z., & Sulaiman, M. H. (2023). Stock price predictive analysis: an application of hybrid barnacles mating optimizer with an artificial neural network. *International Journal of Cognitive Computing in Engineering, 4*. https://doi.org/10.1016/j.ijcce.2023.03.003

Neelakandan, S., & Paulraj, D. (2020). A gradient-boosted decision tree-based sentiment classification of Twitter data. *International Journal of Wavelets, Multiresolution, and Information Processing, 18*(04), Article 2050027.

Nilabar Nisha, U., Manikandan, A., Venkataramanan, C., & Dhanapal, R. (2023). A score based link delay aware routing protocol to improve energy optimization in wireless sensor network. *Journal of Engineering Research*. https://doi.org/10.1016/j.jer.2023.100115

Palaniappan, M., & Annamalai, M. (2019). *Advances in signal and image processing in biomedical applications*. https://doi.org/10.5772/intechopen.88759

Patel, R., & Passi, K. (2020). Sentiment analysis on Twitter data of the World Cup soccer tournament using machine learning. *IoT, 1*(2), 218–239.

Patra, B. G., Das, D., Das, A., & Prasath, R. (2015). Shared task on sentiment analysis in Indian languages (sail) tweets—An overview. *In Mike*, 650–655.

Phan, H. T., Tran, V. C., Nguyen, N. T., & Hwang, D. (2020). Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. *IEEE Access : Practical Innovations, Open Solutions, 8*, 14630–14641.

Reka, R., Manikandan, A., Venkataramanan, C., et al. (2023). An energy efficient clustering with enhanced chicken swarm optimization algorithm with adaptive position routing protocol in mobile adhoc network. *Telecommunication Systems*. https://doi.org/10.1007/s11235-023-01041-

Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., & Rodriguez, D. Z. (2019). A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics, 15*(4), 2124–2135.

Tam, S., Said, R. B., & Tanriover, O. O (2021). A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification. *IEEE Access : Practical Innovations, Open Solutions, 9*, 41283–41293.

Venkataramanan, C., Ramalingam, S., & Manikandan, A (2021). LWBA: Lévy-walk bat algorithm based data prediction for precision agriculture in wireless sensor networks. *Journal of Intelligent & Fuzzy Systems, 41*, 2891–2904. https://doi.org/10.3233/JIFS-202953

Wang, G., Lu, M., Dong, Y. Q., & Zhao, X. J. (2016). Self-adaptive extreme learning machine. *Neural Computing and Applications, 27*(2), 291–303.

Yi, J., Wang, J., & Wang, G. (2016). Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problems. *Advances in Mechanical Engineering, 8*(1), Article 168781401562483.

Zuo, E., Zhao, H., Chen, B., & Chen, Q. (2020). Context-specific heterogeneous graph convolutional network for implicit sentiment analysis. *IEEE Access : Practical Innovations, Open Solutions, 8*, 37967–37975.