Transformer-based Generative Adversarial Network (GAN) hybrid model for sentiment analysis

hridesh kumar1, Dr.dayashankar Pandey2

hrideshkumar111@gmail.com1,dayashankar.rkdfist@gmail.com2

1M Tech Scholar Dept.of Information Technology , RKDF IST SRK UNIVERSITY, BHOPAL

2 HOD, Dept.of Information Technology, RKDF IST SRK UNIVERSITY, BHOPAL

Abstract

Sentiment analysis, which focuses on extracting insights from textual data to reveal people's opinions or attitudes on a specific issue, has become a prominent area of research in natural language processing (NLP), especially with the rise of social media. Twitter, as a popular platform, allows users to openly share their views and thoughts. However, analyzing Twitter data presents unique challenges due to the frequent use of slang, abbreviations, and misspellings in its short-form content. Traditional automated feature selection methods face limitations, including increased computational costs as the number of features grows. Deep learning, with its ability to self-learn and efficiently process large datasets, helps address these challenges. This paper proposes using a conditional generative adversarial network (GAN) for sentiment analysis on Twitter, with a convolutional neural network (CNN) employed to extract features from the data. Compared to previous approaches, the proposed method achieves superior results in terms of accuracy, recall, precision, and F1 score, with a classification accuracy of 93.33%.

Introduction

The rapid expansion and widespread availability of the internet have encouraged individuals to express their opinions and viewpoints on social media platforms. Today, social media serves as a primary channel for disseminating information across society [1]. Platforms like Facebook, Twitter, Instagram, and LinkedIn have attracted billions of users, drawn to the digital landscape. Twitter, a well-known microblogging service, allows users to post short, real-time messages called tweets, limited to 140 characters, to share their thoughts, views, and ideas [2]. Researchers have analyzed Twitter data for a variety of purposes, including sentiment analysis, public health surveillance, political trends, education, and sports [3].

Twitter sentiment analysis (TSA) seeks to understand how users feel about various topics, people, events, services, products, and organizations based on the content of their tweets. The platform's success has contributed to the growing prominence of sentiment analysis, which is typically approached using three main techniques: statistical methods, lexicon-based approaches (knowledge-based methods), and hybrid strategies [4]. Lexicon-based approaches are efficient and scalable but may struggle to accurately detect emotions due to language complexities [5]. Traditional sentiment analysis techniques often rely on shallow models trained on carefully selected features, using methods such as n-grams, part-of-speech (POS) tagging, and classifiers like support vector machines (SVM), latent Dirichlet allocation (LDA), and Naive Bayes [6]. However, these techniques often face challenges in achieving accurate classification results due to the complexity of feature engineering, which heavily influences the outcomes [7]. Deep learning has been explored to address this, with applications in areas such as personality analysis, online social network age group categorization, and sentiment analysis ([8].

Since the advent of deep learning, sentiment analysis has seen significant advancements. Deep learning algorithms, which consist of multiple processing layers, enable the learning of diverse data representations by capturing various levels of abstraction. As noted by [11], these algorithms can automatically extract important features from data without requiring manual intervention. One of the most widely used techniques for word vector encoding in this context is word embedding, which effectively preserves both semantic and syntactic information [12].

While deep learning models typically require a large amount of labeled data, which can be time-consuming and costly to collect, generative networks have proven useful in addressing this challenge. Generative Adversarial Networks (GANs), in particular, have enhanced classification accuracy [9]. GANs offer a novel approach to training generative models by framing the problem as a supervised learning task. These networks can automatically recognize patterns, with the GAN consisting of two sub-models: the generator, which creates new data samples, and the discriminator, which distinguishes between real and generated samples [10].

The key contributions of this paper are as follows:

- A deep learning framework for Twitter sentiment analysis using a Conditional GAN (CGAN/GAN) is proposed.
- Features from Twitter data are extracted using a convolutional neural network (CNN).
- A Long Short-Term Memory (LSTM) network is integrated into the discriminator to improve classification accuracy.

2. Literature Review

Social media is an integral part of modern life, and its data provides valuable insights for various studies. Several approaches have been developed for sentiment analysis using social media data. For instance, a lexicon-based method using WordNet or SentiWordNet can identify emotional words in context and assign sentiment polarity (Gopalan et al., 2023). Gupta et al. (2021) developed a prediction model to assess public awareness of safety measures in key regions of Saudi Arabia, while Venkataramanan et al. (2021) proposed a multilingual Twitter sentiment analysis (MLTSA) method. The MLTSA approach addresses two key challenges: identifying and translating non-English tweets into English, and applying natural language processing (NLP) techniques to reduce data sparsity.

Hassonah et al. (2020) introduced a novel feature reduction technique using a genetic algorithm (GA), which reduces the size of the feature set by up to 42% without sacrificing accuracy. They also proposed using a specific geopolitical location as a case study for sentiment analysis. Iqbal et al. (2019) explored different sentiment analysis feature sets and classifiers, comparing modern deep learning, ensemble-based techniques, and traditional machine learning approaches. Their results showed that classifier performance is significantly influenced by the choice of feature sets.

In another study, Jin et al. (2020) analyzed public sentiment toward India's COVID-19 lockdown using NLP and machine learning classifiers. Neelakandan and Paulraj (2020) proposed an LSTM-based recurrent neural network (RNN) for sentiment analysis, capable of representing multiple emotions through multi-label classification. Manikandan et al. (2023) enhanced the Bi-directional LSTM (Bi-LSTM) model by using pre-trained word embeddings to reduce text representation size and avoid data sparsity. They also implemented an attention mechanism to prioritize important contextual information.

Patel and Passi (2020) introduced an ensemble classifier that combines multiple weak base classifiers, improving sentiment classification performance on standard datasets. Patra et al. (2015) developed a method to classify extremist content on Twitter. Phan et al. (2020) proposed a lexicon-enhanced LSTM model that uses a sentiment lexicon as supplementary input to enhance word sentiment classification.

Chandramohan et al. (2023) suggested self-adaptive probabilistic neural networks (PNNs) to address various problem sizes. They utilized the Windowed Multivariate Autoregressive Model (WMAR) and Independent Component Analysis (ICA) for data preprocessing. Tam et al. (2021) introduced the self-adaptive extreme learning machine (SaELM), an enhanced version of the Extreme Learning Machine (ELM), which optimizes the number of neurons without needing parameter adjustments during training.

Wang et al. (2016) proposed a new collaborative filtering strategy for recommendation systems, incorporating cognitive factors like decision-making styles. Their Efficient Gowers-Jaccard-Sigmoid Measure (EGJSM) combines the Jaccard and Gower coefficients with a nonlinear sigmoid function, improving performance on benchmark datasets (Reka et al., 2023). Additionally, Nilabar Nisha et al. (2023) proposed a deep learning approach for detecting malware variants, transforming malicious code into grayscale images for classification using CNNs, which automatically extract image features.

Mustaffa and Sulaiman (2023) applied the Barnacles Mating Optimizer (BMO), a new optimization algorithm, to solve multi-class sentiment analysis problems in computational linguistics. Haque et al. (2023) highlighted the increasing impact of fake news and proposed a model for fake news detection based on a BERT model connected to an LSTM layer. Lastly, Demircan et al. (2021) focused on determining social media sentiments using machine learning techniques.

3. Proposed Architecture

This research focuses on analyzing the sentiment of tweets, categorizing them as positive or negative. The Conditional Generative Adversarial Network (CGAN/GAN) was employed to classify tweets based on sentiment scores. The overall architecture used in the sentiment analysis process is depicted in Figure 1, which outlines several key stages.

The flowchart showcases the data preprocessing phase, where tweets are cleaned using tokenization, removal of stop words, hashtags, usernames, punctuation, as well as stemming and lemmatization techniques. Following this, features are extracted, and the CGAN/GAN model is trained to predict whether a tweet is positive or negative.

3.1 Dataset

The dataset used in this study is the "US Election 2020 Tweets" dataset, which contains information such as tweet ID, content, retweet count, etc. Table 1 presents the details of the dataset's features.

3.2 Data Preprocessing

Tweets often contain redundant and irrelevant data that needs to be cleaned before analysis. Several preprocessing techniques were applied to sanitize the tweets, making them suitable for input into the model. The techniques used in this study include:

3.2.1 Tokenization

This process splits a long text into smaller units, called tokens. Since tweets often contain hashtags, emojis, and other symbols with varying meanings, tokenization helps break down each element into manageable parts. Example: ['@user', 'I', 'wonder', 'what', 'drugs', '#Trump', 'takes?']

3.2.2 Stop Words Removal

Common words like conjunctions, prepositions, and articles that do not add significant meaning to a sentence were removed to focus on the key content.

3.2.3 Hashtags and Username Removal

Hashtags (denoted by #) and usernames (denoted by @) were excluded from the analysis as they often do not contribute to sentiment classification.

3.2.4 Punctuation Removal

Punctuation marks, which often do not add value in sentiment analysis, were removed to ensure uniformity in the text.

3.2.5 Stemming

Words were reduced to their base forms by stripping suffixes and prefixes. For example, "singing" and "sings" were both converted to "sing" to improve the feature extraction process.

3.2.6 Lemmatization

Lemmatization refines words into their meaningful base forms, considering their context in the sentence. Unlike stemming, it retains the word's true meaning, making the text more context-aware.

3.3 Sentiment Score Calculation

The sentiment score was computed by counting the positive and negative words in each tweet. A score of 1 was assigned for positive tweets, while a score of -1 indicated a negative tweet. The score was then used as input to the CGAN/GAN model for further classification.

3.4 Feature Extraction with CNN

A Convolutional Neural Network (CNN) was applied to extract essential features from the input data. The CNN identified patterns in word similarity by convolving a set of filters over the input, using a ReLU activation function to generate new features. The filters could adjust their parameters to better capture relevant tweet features. Equation (1) represents the convolution operation:

$$s_i = f(v \cdot [w_i : w_i + h - 1] + b)$$

where f is a nonlinear activation function, and v represents the filter.

3.5 CGAN/GAN-based Sentiment Classification

The CGAN/GAN model works similarly to a traditional Generative Adversarial Network (GAN), with the addition of an input condition yyy. This condition is provided to both the generator and discriminator, helping the CGAN/GAN model generate and classify fake samples based on the sentiment score of the input tweets. The objective function used in the CGAN/GAN is:

$$\min_G \max_D L(G, D) = E_{s \sim q_{data}}[\log D(s|r)] + E_{z \sim q_z}[\log(1 - D(G(z|r)))]$$

3.6 Generator

The generator GGG creates synthetic Twitter samples, which the discriminator DDD then classifies as real or fake. This adversarial setup pushes both models to improve. The generator's loss function seeks to fool the discriminator into accepting the synthetic samples as real, while the discriminator strives to correctly classify the samples as fake or real.

3.7 Discriminator with LSTM

The discriminator incorporates a Long Short-Term Memory (LSTM) network to capture both short-term and long-term dependencies in the tweet sequences. LSTM units contain forget, update, and output gates, each performing specific roles in determining what information to retain or discard.

$$F_t = \sigma(W_f[y_{t-1}, X_t] + b_f)$$

$$U_t = \sigma(W_u[y_{t-1}, X_t] + b_u)$$

The LSTM enhances the discriminator's ability to classify complex tweet sequences more accurately.

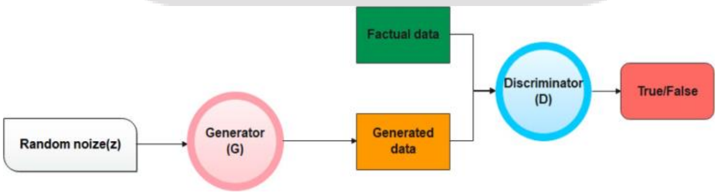

Fig. 2. Basic structure of GAN.

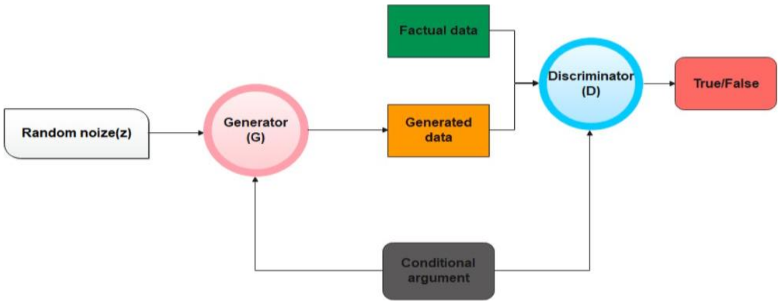

Fig. 3. Structure of Proposed GAN.

4. Results and discussion

This section outlines the sentiment analysis findings using the suggested CGAN/GAN 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.

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

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.

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

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

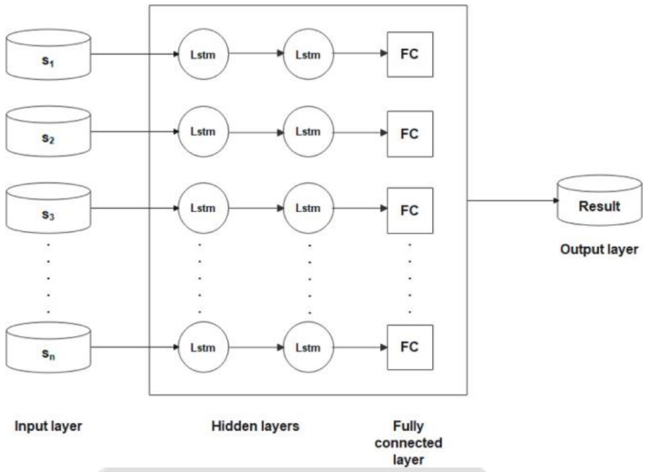

Fig. 4. Structure of discriminator.

One limitation of current algorithms is their inability to fully capture relationships between input features. For example, while CNN focuses on local patterns, it struggles to identify long-term dependencies. LSTM and Bi-LSTM are better at handling these long-term dependencies but often fall short in modeling complex interactions between input features. The proposed CGAN/GAN approach addresses this issue by employing a GAN to learn the data distribution and generate synthetic samples. This method improves classification accuracy by capturing correlations between input features and enhancing the training process with these synthetic samples. Additionally, GANs allow for the creation of artificial data that closely resembles the original, which is valuable when real data is limited or difficult to obtain.

As seen in Table 4 and Figure 6, the proposed CGAN/GAN model demonstrates superior precision compared to existing methods. The results show that CGAN/GAN achieves an accuracy rate of 92.86%, surpassing other algorithms. CNN, for example, has the lowest precision at 85.98%, as it is more suited for tasks involving spatial information like image recognition rather than sequential data like text. LSTM and Bi-LSTM perform better, with precision rates of 91.45% and 91.55%, respectively, though LSTM faces challenges with long-term dependencies due to the vanishing gradient problem. Bi-LSTM mitigates this issue by processing data bidirectionally but requires more computational resources.

Table 5 and Figure 7 reveal that CGAN/GAN outperforms other algorithms in terms of recall, achieving a high score of 93.11%. In contrast, CNN, LSTM, and Bi-LSTM have lower recall scores ranging from 87.63% to 90.29%, indicating these models might miss relevant data points. The recall score of 93.11% suggests that CGAN/GAN is more effective in identifying key data points, making it advantageous for tasks that require comprehensive and accurate data analysis. The ability of CGAN/GAN to generate synthetic data also adds value by filling data gaps or providing additional context.

Furthermore, Figure 7 illustrates that CGAN/GAN achieved the highest F1 score of 92.68%, outperforming CNN, LSTM, and Bi-LSTM. LSTM followed with a score of 85.47%, while CNN scored 86.6%. This indicates that CGAN/GAN excelled in classifying Twitter data, with its high F1 score demonstrating its ability to both accurately classify data and generate realistic synthetic data. Lastly, as shown in Table 7, CGAN/GAN requires less processing time compared to other classifiers when analyzing the same number of tweets, further highlighting its efficiency in performance.

Table 3

Accuracy comparison of proposed work with the existing work.

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

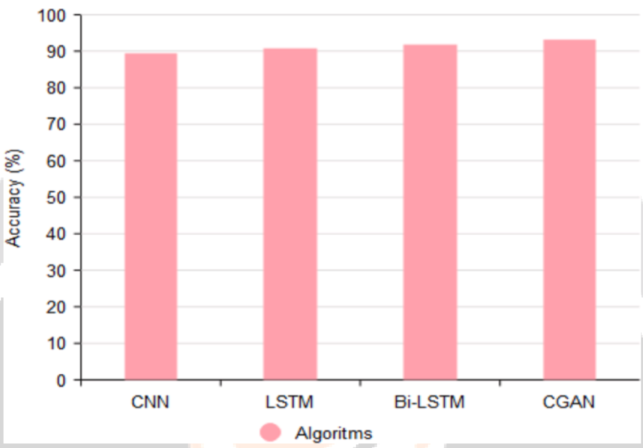

Fig. 5. Comparison of the accuracy of the proposed algorithm with the existing algorithm.

Table 4
Comparison of precision between proposed and existing algorithms.

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

Table 7 and Figure 8 provide an analysis of CNN, LSTM, and Bi-LSTM in relation to the number of tweets and the time required for processing. Figure 8 further details the performance of the CGAN/GAN classifier regarding the number of tweets and processing time. The results show the time (in seconds) each classifier takes to analyze Twitter data for sample sizes of 500, 1000, 1500, 2000, and 2500.

CNN consistently requires the longest processing time. For 500 tweets, CNN takes 0.056 seconds; for 1000 tweets, 0.072 seconds; for 1500 tweets, 0.12 seconds; for 2000 tweets, 0.138 seconds; and for 2500 tweets, 0.189 seconds. As the sample size grows, CNN's processing time increases significantly.

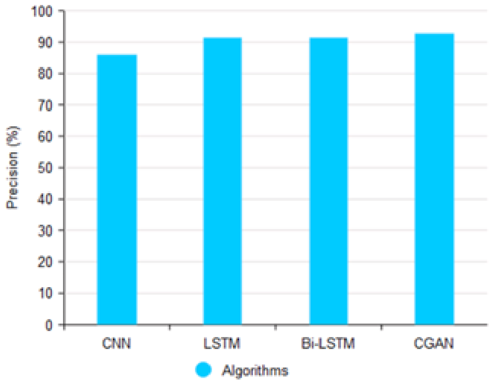

Fig. 6. shows the precision comparison between the proposed and existing algorithms.

Table 5

Illustrates the recall comparison between the pro- posed algorithm and the existing algorithm.

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

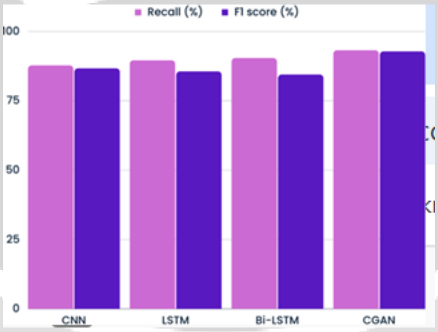

Fig. 7. Recall & F1-Score Comparison between the Proposed and Existing Algorithms.

In analyzing Twitter data, CGAN/GAN is the fastest classifier, while CNN takes the longest processing time. LSTM and Bi-LSTM run almost concurrently, with both taking more time than CGAN/GAN but less than CNN. These results suggest that CNN is the least efficient classifier in terms of processing speed, while CGAN/GAN, LSTM, and Bi-LSTM are the most efficient for Twitter data analysis.

Table 8 outlines the key hyperparameters for LSTM, highlighting the importance of tuning hyperparameters to achieve optimal model performance. The hyperparameters were fine-tuned using the randomized search method to maximize accuracy. The recommended hyperparameters for the proposed sentiment analysis model include the learning rate, hidden layers, activation function, dropout rate, batch size, and number of epochs.

Table 6

Comparison of F1 score between proposed and exist- ing algorithms.

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

Specific hyperparameters play a critical role in optimizing neural network-based sentiment analysis models. The learning rate, which controls the step size in each optimization iteration, is set at 0.01—a moderate value that allows the model to converge quickly and avoid local minima. Hidden layers, representing different processing levels within the network, are set at 2, which is suitable for sentiment analysis. While adding more hidden layers can enhance the model's ability to learn complex patterns, it also increases the risk of overfitting.

The Rectified Linear Unit (ReLU) activation function is used to introduce non-linearity into the network. It is a simple yet effective function that sets negative values to zero and has proven highly effective in various natural language processing (NLP) tasks. To further combat overfitting, a dropout rate of 0.5 is applied, meaning half of the neurons are randomly deactivated during training, helping the model generalize better.

The batch size, which determines the number of samples processed in each training iteration, is set at 128, striking a balance between training speed and model performance. The number of epochs, or complete passes through the training data, is set at 30, providing sufficient training without overfitting the model.

The key hyperparameters for the proposed sentiment analysis model are:

- Two hidden layers.
- A moderate learning rate of 0.01.
- A 0.5 dropout rate.
- ReLU activation function.
- 30 epochs.
- Batch size of 128.

These hyperparameters help the model effectively learn complex patterns while minimizing the risk of overfitting.

Conclusion

Twitter is one of the most popular platforms for sharing information. This paper introduces a CGAN/GAN model for sentiment analysis on Twitter. The primary contribution is integrating a CNN in the GAN generator and using an LSTM in the GAN discriminator. For optimal performance of the neural network-based sentiment analysis models, several key hyperparameters must be configured. One critical hyperparameter is the learning rate, which controls the step size during each optimization iteration. The key hyperparameters used in the proposed sentiment analysis model include:

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

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/GAN	0.035	0.058	0.069	0.117	0.153

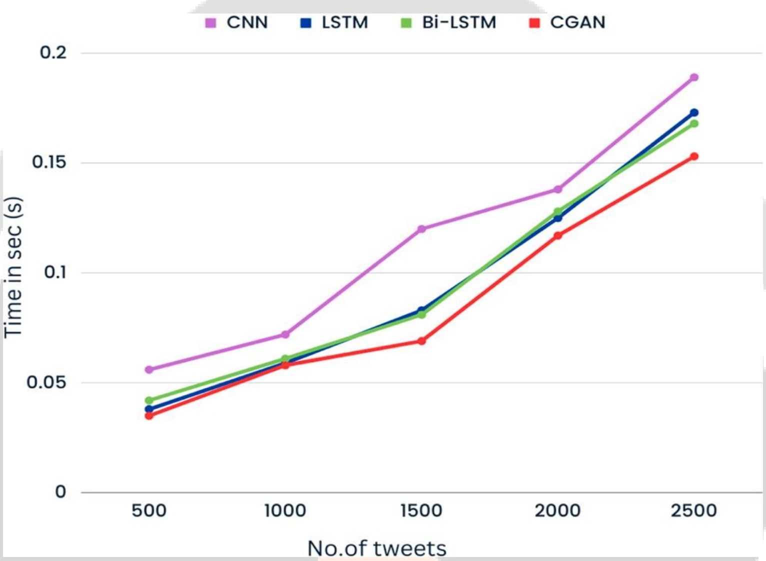

Fig. 8. Shows the Time consumption for several tweets.

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

Using these hyperparameters, the model effectively captures complex patterns while balancing overfitting and learning. The CGAN/GAN achieves an accuracy of 93.33%, significantly higher than other approaches. An analysis of Twitter data reveals that CGAN/GAN is the fastest classifier, while CNN takes the most processing time. LSTM and Bi-LSTM have similar runtimes, both slower than CGAN/GAN but faster than CNN. CGAN/GAN processes Twitter data quickly: 0.035 seconds for 500 samples, 0.058 seconds for 1000 samples, 0.069 seconds for 1500 samples, 0.117 seconds for 2000 samples, and 0.153 seconds for 2500 samples. By incorporating synthetic data in the training process, CGAN/GAN enhances accuracy by better identifying relationships between input features.

REFRENCES

[1]. 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

[2]. Ahmad, S., Asghar, M. Z., Alotaibi, F. M., & Awan, I. (2019). Detection and classification of social media-based extremist affiliations using sentiment analysis techniques.

[3]. Human-Centric Computing and Information Sciences, 9(1).

[4]. 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.

[5]. 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.

[6]. 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

[7]. 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.

[8]. Bommaraju, K., Manikandan, A., & Ramalingam, S. (2017). Aided system for visually impaired people in bus transport using intel galileo gen-2: technical note.

[9]. International Journal of Vehicle Structures and Systems, 9(2), 110–112. https://doi. org/10.4273/ijvss.9.2.09

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

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

[12]. Chandramohan, K., Manikandan, A., Ramalingam, S., & Dhanapal, R. (2023).

[13]. 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

[14]. 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.

[15]. 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

[16]. 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.

[17]. 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

[18]. 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.

[19]. 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

[20]. 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.

[21]. 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.

[22]. 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.

[23]. 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

[24]. 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

[25]. 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

[26]. 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.

[27]. 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

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

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

[30]. 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.

[31]. 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.

[32]. 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-

[33]. 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.

[34]. 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.

[35]. Venkataramanan, C., Ramalingam, S., & Manikandan, A (2021). LWBA: L´evy-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

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

[37]. 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.

[38]. 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.