Sentiment Analysis of English Texts Using Deep Learning Methods: Comparative Study

Aseel Qararia
College of Information Technology
Palestine Technical University - Kadoorie
Tulkarem, Palestine
aseelsameerqararia@gmail.com

Mahmmoud Jazzar
College of Information Technology
Palestine Technical University - Kadoorie
Tulkarem, Palestine
m.jazzar@ptuk.edu.ps

Amna Eleyan
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester-UK
a.eleyan@mmu.ac.uk

Tarek Bejaoui
Computer Engineering Department
University of Carthage
Tunisia
tarek.bejaoui@ieee.org

***Abstract*—With the great development of social media and people's increasing reliance on it, these platforms produce huge amounts of data daily, which reflect users' opinions and interactions in various fields. In this research, we chose Twitter for the study, as it is one of the most prominent platforms that allow opinions to be freely and directly expressed. Due to the huge volume of data published on Twitter, analyzing and understanding it becomes a difficult task without relying on advanced techniques such as deep learning (DL). Therefore, this study came to compare several popular methods in analyzing tweet sentiment while evaluating their performance using different criteria: accuracy, error rate, precision, recall, F1 score, and training time. In this study, we used the most common methods in sentiment analysis, namely Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). The results showed that BERT is the best method for this task, achieving 91% accuracy and a 9% error rate, but its training time was much longer compared to other methods. Although BERT is the best choice, RNN or CNN are better alternatives when speed is a priority and resources are limited.**

***Keywords*—Social media, Twitter, Sentiment analysis, DL, RNN, CNN, LSTM, BERT, Performance evaluation, and Resource limitations**

I. INTRODUCTION

With the proliferation of social media and rapid technological development, platforms such as Twitter have become a primary means for millions to express their opinions on various topics, such as politics, personal affairs, business, and others. This great diversity in users and age groups has led to the production of a huge amount of published daily data. Therefore, there is an urgent need to develop technologies capable of analyzing this data and understanding the public sentiments it reflects, which enables organizations to improve their strategies based on these analyses [1], [16].

Sentiment analysis (SA) is an important field of natural language processing (NLP). It is used to understand and analyze texts and automatically classify sentiments as positive, negative, or neutral. This analysis helps to access users' feelings and opinions through the words they write in their tweets, which contributes to a better understanding of their sentiments and opinions [2].

Due to the huge amount and rapid growth of data, there is an urgent need to use DL methods to analyze these texts and improve the accuracy of the results. With many DL methods available, it is important to understand the performance of each method when applied to this data and to know its strengths and weaknesses, which requires careful evaluation and study [3].

In this paper, we present a comparative study of DL methods for SA on Twitter tweets. The methods included in this study are CNN, RNN, LSTM, and BERT. These methods were chosen based on a review of prior research demonstrating their effectiveness in SA tasks. Jin et al. [4] proved that CNN, RNN, and LSTM are powerful tools for SA because of their ability to recognize complex patterns in texts. On the other hand, Colón et al. [5] confirmed that BERT is one of the best methods for analyzing complex texts because of its high ability to understand subtle meanings in texts.

These methods were trained on a dataset containing tweets from Twitter, and their performance was evaluated based on multiple criteria. The goal of this comparison is to determine the most efficient and suitable method for analyzing the sentiment of English texts, which will contribute to the development and improvement of automated SA techniques in the future.

II. BACKGROUND

A. Twitter Sentiment Analysis

SA is one of the main areas of NLP [2], which focuses on analyzing and classifying text sentiments. One of its important applications is text analysis in social media, which has become a major platform for individuals to express their opinions, whether personal, political, or evaluating products and services [1].

This process aims to extract the emotional tone from texts, whether the sentiments are positive, negative, or neutral, to reach results that contribute to the development of services or company products or even follow the trends of users in a specific area to direct progress in any field based on these results. This approach provides a deeper understanding of public opinion [3].

Due to the huge amount of data flowing through these platforms, it is difficult to track and analyze them manually, which makes artificial intelligence techniques necessary to understand users' trends, especially on platforms such as Twitter, where SA is used to extract patterns and opinions from published tweets [2].

Texts are classified in SA into three main levels: "document level, sentence level, and feature level". In the case of Twitter tweets, classification is usually done at the sentence level, because tweets are short and are considered closer to single sentences than documents that usually consist of several connected sentences. In addition, this type of classification focuses on identifying the general sentiment associated with each tweet, without going into depth to analyze the specific features within the text. Accordingly, sentence classification is the most suitable approach for classifying the sentiment of tweets [6].

For sentiment analysis, there are three approaches as shown in Figure 1. In the case of Twitter tweet analysis, machine learning (ML), especially DL, is considered the most suitable, because it has the ability to deal with complex texts that contain large data, which makes it superior to lexicon approaches. In addition, DL has a high ability to adapt to large and unstructured data, such as tweets that contain abbreviations, symbols, hashtags, etc., which makes it more efficient compared to hybrid approaches. And let's not forget the accuracy, as DL methods have proven their high ability to analyze sentiment more accurately compared to other methods [7].

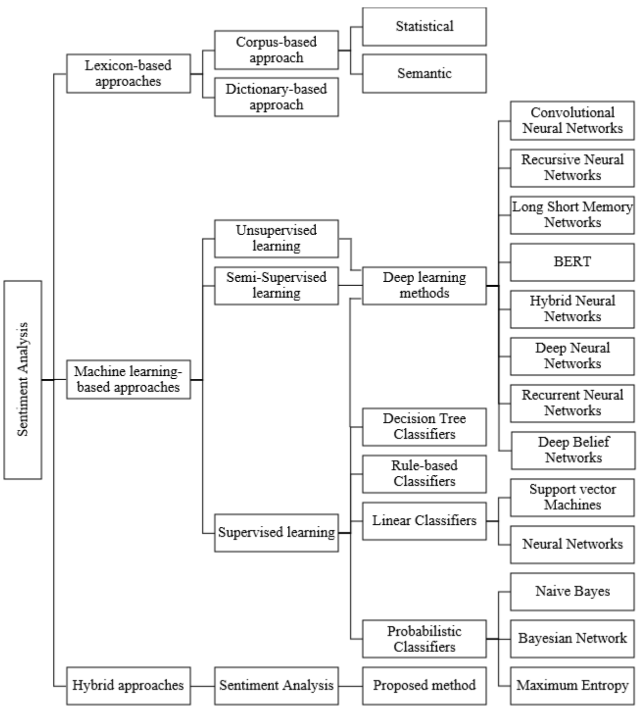

Figure 1. Sentiment Analysis Approaches

B. Deep Learning Methods

It is a field of ML characterized by its ability to extract patterns and learn from data by using multi-layer artificial neural networks. These networks simulate the work of the human brain in making decisions and processing data, which distinguishes them and gives them speed and accuracy in performance [7]. The development of this method, their excellence in performance, and their ability to deal with textual representations, especially in complex texts, made them an effective method in NLP tasks, including SA [2].

It is characterized by their ability to deal with the challenges of textual data such as synonymy, implicit sentiment, and writing differences. For example, in English texts, there is a difference between the American and British dialects, and this difference has caused a problem in analysis when using old analysis methods because they treat words in isolation from their position within the sentence. If we look at the word "color" and the word "colour", they both have the same meaning, but they were treated as two different words. In addition, the old methods had difficulty in understanding complex linguistic expressions; for example, in the following sentence, "I wouldn't say that the movie was entirely bad", the sentiment of the text must be determined based on the context of the entire sentence and not just the word "bad". DL methods care about the sequence of words and understand the correct meaning of the presence of any word in the sentence, so they understand deep sentiments that are not limited to the presence of a single word (e.g., bad) in the sentence [2, 8].

Since the textual data we are dealing with here are tweets from Twitter users, we need methods that are able to understand the context of the sentence and deal with implicit sentiments and synonyms in addition to emojis, symbols, etc., so we chose this approach to use in SA because of its ability to deal with fine linguistic details and understand sentiments in various contexts with different dialects and languages.

Figure 2 shows the subcategories of DL methods. In SA of textual data, we use supervised methods because they rely on clearly labeled data, so they are better suited for tasks that require accurate classification of sentiments, unlike unsupervised methods, which are used for tasks of discovering patterns in data and hidden relationships in it. As for reinforcement learning, it is a category of DL and relies on trial and error mechanisms only [9].

In this comparison, we will use CNN, RNN, LSTM, and BERT because they have shown effectiveness in analyzing texts, accurately identifying sentiments, and understanding complex texts in previous studies and experiments [4, 5].

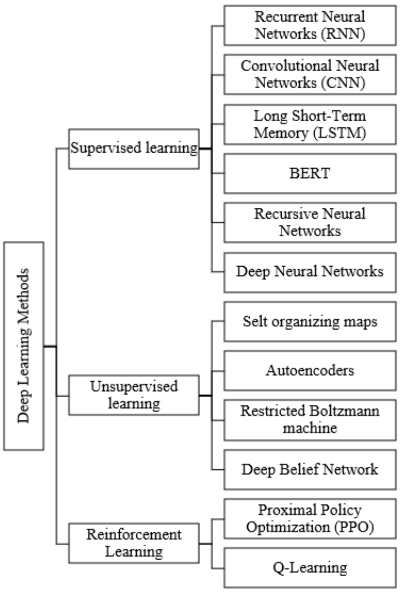

Figure 2. Deep Learning Methods

Let us now discuss in detail the methods used in the comparison process.

C. CNN Method

A CNN method is a special type of deep neural network designed to process images but has been developed to process text and audio as well. It is characterized by its ability to extract features from data automatically, which gives it good performance and satisfactory results. This is due to a set of characteristics that distinguish it, as the nodes in it are connected to a small part of the nodes in the previous layers, which helped reduce the number of transactions and improve performance. It also uses filters (kernels) to discover different data patterns. In addition, it is applicable to texts of variable length [10].

The CNN method is used to extract linguistic patterns, contexts, and other basic features in texts. It relies on converting words into numerical representations using word embeddings, which helps in reaching semantic relationships between words. It is also effective in processing high-dimensional texts, as it is distinguished from traditional methods because it reduces the need for data preparation steps [11].

D. RNN Method

RNNs are a type of deep neural network used in processing sequential data because they form their own connections between neurons, which creates feedback loops within the algorithm, as this algorithm uses sequential temporal information by storing an internal state that is updated at each step, making it suitable for understanding the sequence in the data [7].

Since this algorithm is designed to deal with text data sequences, it is suitable for dealing with text sequences that contain multiple dialects, hashtags, and emojis. The algorithm focuses on the important parts of the text, which enhances the accuracy of classifying sentiments in diverse texts [12].

E. LSTM Method

The LSTM method is one of the deep neural networks that were developed to predict the behavior of non-linear sequential text data. It is designed to access contextual information and long-term dependencies [13]. It is distinguished in its role in SA for its ability to deal with complex texts and deeply comprehend and understand texts accurately. It is also able to understand the temporal context of texts in addition to the dependencies that it can access. Its layers consist of memory units that are repeatedly connected to each other. Each unit contains three multiple gates: the input, output, and forget gate. These gates allow information to be used, stored, and forgotten for long periods of time [5].

F. BERT Method

It is a modern and innovative model in NLP and is one of the most prominent developments that have contributed to analyzing texts with high accuracy, including sentiment analysis[17]. What distinguishes BERT is its ability to understand the context of words from both sides (before and after the word), making it more accurate compared to traditional methods. BERT is based on the Transformers structure, which gives it high efficiency in dealing with texts [18]. The model has been pre-trained on a large and diverse set of data, which helps it understand general linguistic patterns. In addition, the model can be easily customized for specific tasks using the fine-tuning technique by quickly training on small data specific to the required task, such as analyzing the sentiment of tweets or product reviews. One of its most prominent features is its support for many languages, as ready-made models are available and trained in different languages, making it suitable for use in global applications. Although its large size and long training time may seem challenging, its high accuracy in understanding and analyzing texts compensates for that, making it an ideal choice for processing complex texts [14, 15].

In the field of sentiment analysis, BERT has proven effective in improving results and reducing errors, helping companies understand customer sentiment more deeply and use these insights to make informed strategic decisions. BERT has become a modern and advanced standard in text analysis, enhancing machines’ understanding of human language and improving applications in multiple fields [15].

III. Methodology

A. Dataset

We used a subset of the “Twitter Sentiment Analysis Dataset” prepared by Sherif Hussein from Mansoura University in Egypt, available on the Mendeley Data platform.

This dataset consists of 10,000 tweets with few missing values, in addition to two main columns:

- Category: contains the numerical classification tweets: **-1** (negative), **0** (neutral), and **1** (positive).
- Clean_text: contains the tweets' text after preprocessing and cleaning.

This dataset is well-suited for training and evaluating methods due to its diverse distribution of sentiment categories, offering a reliable foundation for comparing the performance of SA techniques on real Twitter data.

B. Data preprocessing

We preprocessed the dataset before starting to train DL methods on it. The processing process was done in these steps:

1) *Removing missing values:* We removed all missing values from the dataset using the "dropna()" function.

2) *Handling Negative Category Values:* The category column contains three categories: -1, 0, and 1, but negative values are not compatible with DL methods, so we changed the value that is represented as negative sentiment from -1 to 2 using the "replace()" function.

3) *Data splitting:* We split the data into training and testing data (80% of the data for training and 20% for testing). These ratios were chosen to balance the training and testing process. The ratio of 80% is enough to train a strong method, and the ratio of 20% is enough to test the trained methods because the data set is large; the ratio provides reliable results to evaluate the performance of each DL method.

4) *Text Tokenization:* Since the dataset contains text data, we converted this data into numeric sequences so that it can be used by DL models. In traditional models like CNN, RNN, and LSTM, we used Tokenizer from Keras library. In BERT model, we used BertTokenizer from Hugging Face library because it provides outputs that fit the nature of the model.

5) *Text padding:* As the texts used are tweets from Twitter users, the lengths of the texts of these tweets are different, so we need to standardize by using padding the length of the input sequences to enable the DL models to process these texts without losing data. In this experiment, the maximum length is 50 characters.

In this study, we used standard preprocessing steps. Future research could explore improvements and other preprocessing strategies, such as stemming, lemmatization, handling of emojis, etc., to improve the performance of SA models.

C. Methods Used

1) *CNN*: In this method, the data is passed through the Conv1D layer, which detects important local patterns in texts. Then the data is passed to the GlobalMaxPooling1D layer, which reduces the dimensionality while preserving the most important features. After that, two dense layers are used: the first is to process the extracted features using the ReLU activation function while the second one classifies the texts into three classes (positive, negative, neutral) using the SoftMax activation function.

2) *RNN:* In this method, words are converted into dense numerical representations using the embedding layer. The data then passes to the SimpleRNN layer, which is a recurrent neural layer that processes the sequential texts and analyzes the words one by one, focusing on the temporal context of the previous words, which enhances the understanding of the relationships between words. Then the data is passed to the GlobalMaxPooling1D layer, which reduces the dimensionality while preserving the most important features. After that, two dense layers are used: the first is to process the extracted features using the ReLU activation function while the second one classifies the texts into three classes (positive, negative, neutral) using the SoftMax activation function.

3) *LSTM*: In this method, words are converted into dense and low-dimensional representations using the embedding layer. Then it passes this data to the LSTM layer, in which the text sequence is processed and the long-term dependencies between words are comprehended, after which the information is either saved or forgotten in the memory cells. Next, a GlobalMaxPooling1D layer is used to reduce complexity and preserve the most important features of the sequence. In the next step, the data is passed through two dense layers: the first one learns the nonlinear relationships in the data using the ReLU activation function, while the second one classifies the texts into three classes (positive, negative, neutral) using the Softmax activation function.

4) *BERT*: In this method, we used the BertForSequenceClassification model to classify texts into three classes. After training the model using the AdamW optimizer algorithm on three training cycles, in each cycle, we calculated the loss and updated the model weights to improve the prediction accuracy.

We trained all methods using the baseline hyperparameter values without any tuning. The number of epochs was set to 3, and the batch size was set to 16 for all methods. We also used the "Adam optimizer" with default settings, and a learning rate of "2e-5" was adopted for the BERT method.

Therefore, we did not optimize the hyperparameter settings in any model, which ensures a fair comparison between the methods, as they are subjected to the same experimental conditions. However, in future work, different hyperparameter configurations will be tested to study their impact on model performance and achieve the best possible results.

D. Criteria used in evaluation

1) *Accuracy:* We used the sklearn.metrics library, which provides us with the accuracy_score() function. To compare the number of correct predictions made by the model and the total number of texts in the test data and return the accuracy ratio achieved by the model.

2) *Error rate:* We used the following equation:

$$\text{Error rate} = 1 - \text{Accuracy}$$

to calculate the error rate, which represents the ratio of the number of incorrect predictions of the model with the total number of texts in the test data. This value is calculated to determine the size of errors made by the model.

3) *Training time:* To calculate the time to train the model on the training dataset, we used the time library to call the time() function to record the time before and after training. After we got them, we applied the following equation:

$$\text{Training time} = \text{Time after training} - \text{Time before training}$$

4) *Classification Report:* We used this report to evaluate models at the category level (positive, negative, and neutral) to get a comprehensive and accurate analysis of model performance. This report includes three criteria:

a) Precision: It expresses the percentage of texts that were correctly classified as belonging to a specific category out of the total texts that were classified as belonging to that category.

b) Recall: It expresses the ability of the model to detect all texts belonging to a specific category.

c) F1-Score: Expresses the balance between precision and recall to avoid bias when there is a difference in the size of the categories.

These criteria are calculated using the classification_report() function from the sklearn.metrics library.

E. Experimental Setup

We used Google Colab to run the code in this experiment. It has an Intel(R) Xeon(R) CPU @ 2.20GHz, this processor has one physical core per processor, with two logical processors (Threads) per physical processor, which operates at a frequency of 2.20 GHz, and has a cache of 56.3 MB. It also provides RAM of up to 12 GB.

IV. Results and Discussions

In this section, we will present and discuss the results of evaluating the performance of DL methods in sentiment analysis. We analyzed the performance of each method under the same conditions according to the following criteria: accuracy, error rate, training time, f1-score, precision, and recall.

Table 1. shows the accuracy and error rate for all methods

Methods	Accuracy	Error Rate
CNN	**%90**	**%10**
RNN	**%90**	**%10**
LSTM	**%87**	**%13**
BERT	**%91**	**%9**

Table 1 presents the accuracy and error rate results for the DL methods evaluated in this study. Among the methods, BERT demonstrated the highest accuracy at 91%, establishing itself as the most effective approach for sentiment classification. This superior performance can be attributed to BERT's utilization of pre-trained language models, which enhance its ability to comprehend text and nuanced language expressions. Both CNN and RNN achieved a comparable accuracy of 90%, indicating their effectiveness in sentiment analysis, though slightly less so than BERT. In contrast, the LSTM algorithm recorded the lowest accuracy at 87%. Regarding error rates, BERT also outperformed the other methods with the lowest error rate of 9%. CNN and RNN followed closely with an equal error rate of 10%, while LSTM had the highest error rate at 13%, reflecting its relative inefficiency in this context.

Table 2. shows the training time for all methods

Methods	Training Time (s)
CNN	**55.24**
RNN	**56.63**
LSTM	**105.77**
BERT	**11684.66**

Table 2 highlights the training time required for each DL method. Among the methods, BERT had the longest training time, taking approximately 3 hours and 15 minutes. This extended duration is attributed to the complexity of the model and the large number of parameters involved. LSTM ranked second in terms of training time, requiring around 106 seconds due to its architecture, which processes long sequences and captures temporal dependencies. In contrast, RNN and CNN demonstrated significantly faster training times, completing the process in approximately 56 and 55 seconds, respectively. This efficiency makes RNN and CNN particularly suitable choices when quick training is a priority.

Table 3. shows the Precision, Recall, and F1-Score for all methods

Methods	Precision	Recall	F1-Score
CNN	**%90**	**%88**	**%89**
RNN	**%91**	**%90**	**%89**
LSTM	**%89**	**%90**	**%86**
BERT	**%91**	**%92**	**%90**

Table 3 presents the classification report results for the DL methods evaluated in this study. Among these methods, BERT demonstrated the best balance across the three metrics (precision, recall, and F1-score) highlighting its strong capability to efficiently handle various categories. Meanwhile, CNN and RNN exhibited comparable performance. However, LSTM showed notable fluctuations in classification performance, indicating potential inconsistencies in its handling of certain categories.

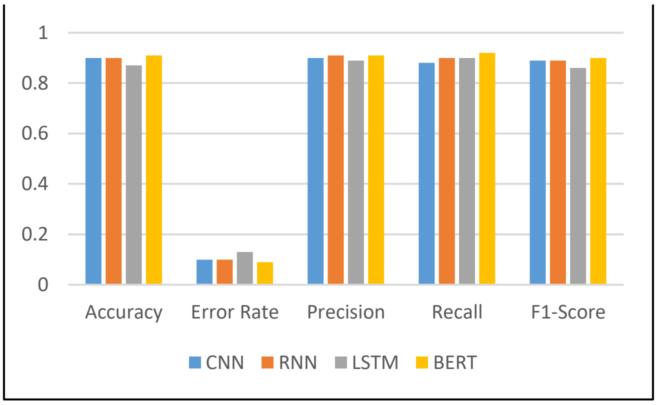

Figure 3. The chart represents the result of the comparison of the DL methods

Based on these results, we conclude that BERT is the most effective method for sentiment analysis, making it the preferred choice when sufficient resources are available and time constraints are not a concern. CNN and RNN, on the other hand, excel in training and execution speed, making them well-suited for applications requiring efficiency and rapid performance. As for the LSTM method, it is essential to carefully evaluate the dataset and its suitability before employing it in SA tasks.

The SA models we've trained are applicable in a variety of practical contexts. They can help organizations understand user sentiment toward specific brands or events. They can also be used to analyze customer comments on social media platforms, helping to uncover interaction patterns and prevailing sentiments. Additionally, these models help track shifts in political discourse and understand public sentiment.

V. Conclusion

In this paper, we compared four deep-learning methods for analyzing the sentiment of English tweets. These methods are CNN, RNN, LSTM, and BERT. The results showed that BERT outperformed other methods, achieving the highest accuracy (91%) and the lowest error rate (9%), indicating its great ability to understand complex texts and fine details. As for the CNN and RNN methods, achieving 90% accuracy proves their high reliability in sentiment analysis. But the LSTM showed the lowest performance with 87% accuracy despite its effectiveness, due to its reliance on sequential processing, which is less efficient in dealing with short texts such as tweets.

By evaluating the classifications, we were able to identify the strengths and weaknesses of the methods used. BERT outperformed in overall classification performance, but the time required to train it is a constraint, especially when resources are limited. In contrast, CNN and RNN methods had a good balance between accuracy and short training times, so they are suitable options when resources are limited. Choosing the best method depends on a good understanding of the task at hand and the nature of the data used, as well as the efficiency of the available resources. Accuracy and processing speed must be balanced to achieve satisfactory results.

In conclusion, we point out that using the BERT method is for tasks that require high accuracy and a deep understanding of the text. But, if the priority of the task is speed and the resources are insufficient to use BERT, CNN and RNN methods are the best choice in this case. Also, future research should focus on improving the training efficiency and using hybrid models to improve the performance of sentiment analysis on the Twitter platform.

Limitations and Future Work:

There are several limitations and work that we recommend and seek to address in future work. In this study, we compared deep learning methods that have proven effective in SA in previous studies, but we did not combine these methods into an improved hybrid model. We believe that building a hybrid model between two or more of the four methods used will improve performance and increase classification accuracy. We also want to use a larger dataset than the one currently used to obtain more accurate and representative results cause Twitter generates massive amounts of data daily. We also want to analyze the impact of computational resources in greater depth to determine the most appropriate model based on the capabilities of the system being used.

References

[1] Rahman, M. S., & Reza, H. (2022). "A Systematic Review towards Big data analytics in social Media". *Big Data Mining and Analytics*, *5*(3), 228–244. https://doi.org/10.26599/bdma.2022.9020009

[2] Tan, K. L., Lee, C. P., & Lim, K. M. (2023). "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research". *Applied Sciences*, *13*(7), 4550. https://doi.org/10.3390/app13074550

[3] Nip, J. Y. M., & Berthelier, B. (2024). "Social Media Sentiment Analysis". *Encyclopedia*, *4*(4), 1590–1598. https://doi.org/10.3390/encyclopedia4040104

[4] Jin, Y., Cheng, K., Wang, X., & Cai, L. (2023). "A Review of Text Sentiment Analysis Methods and Applications". Frontiers in Business Economics and Management, 10(1), 58–64. https://doi.org/10.54097/fbem.v10i1.10171

[5] Colón-Ruiz, C., & Segura-Bedmar, I. (2020). "Comparing deep learning architectures for sentiment analysis on drug reviews". Journal of Biomedical Informatics, 110, 103539. https://doi.org/10.1016/j.jbi.2020.103539

[6] J. F. Raisa, M. Ulfat, A. Al Mueed and S. M. S. Reza, "A Review on Twitter Sentiment Analysis Approaches," 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 2021, pp. 375-379, doi: https://doi.org/10.1109/ICICT4SD50815.2021.9396915

[7] Dang, N. C., Moreno-García, M. N., & De La Prieta, F. (2020). "Sentiment Analysis Based on Deep Learning: A Comparative Study". *Electronics*, *9*(3), 483. https://doi.org/10.3390/electronics9030483

[8] Agüero-Torales, M. M., Salas, J. I. A., & López-Herrera, A. G. (2021). "Deep learning and multilingual sentiment analysis on social media data: An overview". *Applied Soft Computing*, *107*, 107373. https://doi.org/10.1016/j.asoc.2021.107373

[9] Thangarasu, N., & Network, G.A. (2023). "Deep learning algorithms and their relevance: A review". *International Journal of Data Informatics and Intelligent Computing*. https://doi.org/10.59461/ijdiic.v2i4.78

[10] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: https://doi.org/10.1109/TNNLS.2021.3084827

[11] Xie, Y., & C, R. J. R. (2023). "Convolutional Neural Networks for Sentiment Analysis on Weibo Data: A Natural Language Processing Approach". *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2307.06540

[12] Nistor, S. C., Moca, M., Moldovan, D., Oprean, D. B., & Nistor, R. L. (2021). "Building a Twitter Sentiment Analysis System with Recurrent Neural Networks". *Sensors*, *21*(7), 2266. https://doi.org/10.3390/s21072266

[13] Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). "A survey on long short-term memory networks for time series prediction". *Procedia CIRP*, *99*, 650–655. https://doi.org/10.1016/j.procir.2021.03.088

[14] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). "How to Fine-Tune BERT for text classification?" In *Lecture notes in computer science* (pp. 194–206). https://doi.org/10.1007/978-3-030-32381-3_16

[15] Sayeed, M. S., Mohan, V., & Muthu, K. S. (2023). "BERT: A Review of Applications in Sentiment Analysis". *HighTech and Innovation Journal*, *4*(2), 453–462. https://doi.org/10.28991/hij-2023-04-02-015

[16] R. Arul, K. Vishnu, A. Eleyan, and A. K. Bashir, "The authenticity of information on social media," IEEE Technology Policy and Ethics, vol. 5, no. 6, pp. 1–6, 2020.

[17] T. Ali, A. Eleyan, M. Al-Khalidi and T. Bejaoui, T., "GAN-Guarded Fine-Tuning: Enhancing Adversarial Robustness in NLP Models," In *2025 5th IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2025, February, (pp. 1-6). IEEE.

[18] E. M. Damatie, A. Eleyan, and T. Bejaoui, "Real-time email phishing detection using a custom DistilBERT model," in 2024 11th IEEE International Symposium on Networks, Computers and Communications (ISNCC), Washington DC, USA, Oct 2024.