Hybrid CNN and RNN for Twitter Sentiment Analysis

Pratik Pujari[1 (✉)], Adish Padalia[2], Tanish Shah[3] and Dr. Kailas Devadkar[4]

[1]Department of Computer Engineering, Sardar Patel Institute of Technology, University Of Mumbai, Mumbai, India
[2, 3, 4] Department of Information Technology, Sardar Patel Institute of Technology, University Of Mumbai, Mumbai, India
pratikpujari1000@gmail.com

Abstract. Online social media networks have developed into a widespread and significant platform for people to freely express their thoughts and emotions in this era of digital communication. This change in communication habits has important ramifications for brands looking to connect with and understand their target market. Social media is one of the biggest sources of unstructured data, but it takes time to analyze it and extract its meaning. This paper attempts to forecast sentimental analysis based on Twitter user's tweets. We used a Twitter dataset which contains around 520k Tweets, to predict whether the tweet has a positive or a negative connotation which can further help us to understand the mental state of the user. Then, using this dataset, we develop different deep learning models (including RNN and CNN). We examine the outcomes of applying recurrent neural network (RNN) and convolutional neural network (CNN) to these models. Finally, we suggested an approach that combines RNN and CNN to fully exploit each technology's advantages: RNN can learn temporal and context features, notably long-term dependency between multiple entities, while CNN is capable of catching numerous potential features. The result demonstrates that our method is superior to the majority of the existing methods.

Keywords: Sentiment Analysis, Deep Learning, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)

1 Introduction

To assess the user's sentiments, sentiment analysis (SA) employs text analysis, NLP (natural language processing), and statistics. Sentiments are emotions, ideas, or attitudes that are conveyed in relation to a person, circumstance, or object. SA is used to determine whether the data or information obtained is positive or negative. Business experts commonly utilize it to track attitudes, interpret social data, and assess brand reputation and client demands [14,7].The volume of information created or uploaded online has quickly increased as a result of the significant growth in Internet usage in recent years[23,7]. Since the advent of technology, people have utilized social media platforms [18,17] like Twitter, Instagram, Facebook, LinkedIn, YouTube, etc. to communicate their ideas or opinions about various goods, occasions, or objectives. The most popular global microblogging site nowadays for users to express their ideas in the form of quick messages called tweets is Twitter[12].On average, 6,000 tweets are sent on Twitter every second, which equates to more than 350,000 tweets sent every minute, 500 million tweets every day, and around 200 billion tweets every year[9]. Twitter is a reliable OSN (online social network) for user conversations and information sharing. Various elements of our lives are significantly influenced by Twitter mood [11]. In order to extract textual information, SA and text classification further classify the polarity as positive (P) or negative (N). Information may frequently be retrieved from text or tweet content using NLP algorithms. The process by which the machine (computer) determines the meaning of each statement produced by a human is known as NLP-based sentiment classification. TSA (Twitter Sentiment Analysis) manual analysis takes longer and additional professionals are needed for tweet labelling. Due to the availability of big datasets, some of the existing research struggles to attain efficient processing time, complexity, and accuracy. Additionally, the efficiency of the classifier is decreased by the extraction of irrelevant and low-level characteristics. Additionally, the use of all extracted characteristics takes up a lot of space. These flaws render the current algorithms unsuitable for processing data efficiently. These flaws present a research opportunity for an efficient integrated algorithm for the processing of Twitter data. Consequently, an automated model is created to address these issues.

The analysis of online sentiments has made use of the advances of ML (Machine learning) algorithms [22, 10], including SVM (Support Vector Machine), MNB (Multinomial Naive Bayes), LR (Logistic Regression), NB (Naive Bayes), etc. Although these techniques showed good effectiveness, they are relatively slow and require more time to complete the training process. To effectively classify Twitter attitudes, the DL model is introduced. DL is a subclass of ML that employs numerous techniques to resolve challenging issues. DL allows the machine to handle massive amounts of data with little human input by using a series of progressive events. The accurate outcomes of DL-based sentiment analysis may be used for a number of tasks, such as emotion detection [25], product prediction, movie recommendation, and other uses. Several academics have introduced DL in Twitter sentiment analysis as a result of these advancements.

There are numerous issues with employing DL approaches for Twitter sentiment analysis. The author of [6] performed sentiment classification from Twitter data using the DL model. This strategy examined each user's behavioral data to categorize such data. However, this approach has had difficulty extracting precise tweet words from the enormous twitter corpus; as a result, the effectiveness of a classification algorithm has decreased. However, it is unable to lower the dimension that the extracted features take up. As a result, a number of useful traits are inside the local optimum. In [6], they have only used deep convolutional neural networks in order to classify tweet sentiments. The paper proposed a novel CNN architecture which initializes the parameter weights. But it also proposes difficulties like having limited capabilities for word embeddings, difficulty in handling noises and loss in sentiment continuity. In another paper, published by Stanford [16], the neural net structures they experimented included one-hidden-layer Recursive Neural Net (RNN), two-hidden-layer RNN and Recursive Neural Tensor Net (RNTN). RNNs are well suited for handling sequential information processing, handling varying length and effective handling of noise but on the other hand the proposed methodology using RNNs contains a few flaws that are limited context window, long training times, insufficiency in handling short texts, etc. All these flaws can be mitigated by integrating CNNs and RNNs in a hybrid model, one can take advantage of the superior local feature extraction capabilities of CNNs and the sequential learning abilities of RNNs. This hybrid approach can provide a more holistic understanding of sentiment in tweets, effectively addressing the limitations of each individual architecture. Additionally, it can enhance the accuracy of sentiment analysis by fusing both local and contextual information, resulting in more robust and nuanced sentiment predictions in the context of tweet data. With these features, the suggested classifier performed better across the huge dataset and also obtained more accuracy while making less classification errors.

2 Related Work

Sergiu Cosmin Nistor et al [16] developed a sentiment analysis framework using Recurrent Neural Networks for tweet classification, achieving an 80.74% accuracy rate. The method uses an attention mechanism to localize emotion features, enhancing the network's performance. Tested on a large corpus of 1.5 million tweets, the methodology effectively analyses emojis and offers improved results.
In 2022, Rakin Mostafa [15] and his team developed a sentiment analysis method for tweets using Natural Language Processing (NLP) and Bidirectional Long Short Term Memory (Bi-LSTM). The study aimed to classify the types of sentiment in tweets, which was based on how people feel about a company’s products through microblogging. The framework integrated an NLP and Bi-LSTM model to optimally classify the sentiments of the tweets of users.
The study conducted by Bello et al [5] presents a sentiment analysis method using Bidirectional Encoder Representations from Transformers (BERT). The study proposes a text classification method that uses BERT in combination with other variants like CNN, RNN, and BiLSTM. The experimental findings show that these combinations perform well in terms of accuracy rate, precision rate, recall rate, and F1-score compared to when BERT was used with Word2vec and when it was used with no variant.
The authors Ahmad et al [1] used SVM, a widely used supervised machine learning algorithm for textual polarity detection, to analyze the performance of sentiment analysis. The SVM was trained with two pre-classified datasets of tweets. However, the authors also pointed out some problem areas in the training data such as multiple occurrences of tweets, opinion spamming, and dual opinion tweets which could affect the accuracy. But on the contrary, it didn't consider the use of additional features or different feature extraction techniques.

Mandloi and Patel [13] proposed various Machine Learning methods like the Naïve Bayes, SVM, and Maximum Entropy methods are compared. The paper discusses how sentiment analysis is done by these classification algorithms and what is the accuracy and precision in these cases. However, it is important to note that these ML classification models can be effective for sentiment analysis but may not always perfectly capture

the nuances and complexities of human language and sentiment. Also, it is very sensitive to the quality and representativeness of the training data used.
E et al [8], has taken a dataset including US airline online review and discusses the importance of the internet in decision-making, highlighting the need for sentiment analysis to help customers choose the best US airlines. It introduces a new Adaboost approach for sentiment analysis, employing machine learning algorithms for performance analysis. The research aims to bridge gaps between customer views and airlines, potentially claiming it can perform well in other domains as well. Table 1 shows a detailed comparison of discussed related works classified by author, methodology and pros, cons of the proposed model.

Table 1. Comparison of Related works

Author & Year	Methodology	Pros	Cons
Sergiu Cosmin Nistor et al. 2021 [16]	RNN (Recurrent Neural Network)	Can classify tweets with an 80.74% accuracy rate, considering a binary task	Overfitting
Rakin Mostafa et al. 2022 [15]	NLP-BiLSTM (Natural Language Processing and Bidirectional Long Short Term Memory)	Performance of word embedding techniques is good	Lower classification and retrieval accuracy
Abayomi Bello et al. 2023 [5]	BERT model (BERT-CNN, BERT-RNN, BERT-BiLSTM)	Comprehensive approach combining BERT with other techniques	Lower classification accuracy
Munir Ahmad et al. 2017 [1]	SVM (Support Vector Machine)	Able to build a baseline ML model	Not tested on large dataset
Lokesh Mandloi et al. 2020 [13]	Naive Bayes, SVM and Maximum Entropy Method	Naive Bayes as baseline model over SVM	Lower performance
E. Prabhakar et al. 2019 [8]	Improved Adaboost approach	Construct a robust pipeline using basic machine learning models	Lower classification accuracy

3 Methodology

In order to identify the user's sentiment and analyze it effectively, our proposed work makes comparison of three deep learning models. We pre-process the tweets from the dataset efficiently, eventually converting into sequence of integers using tokenizer. The output from the tokenizer is split into train, test datasets and is the input to the model.

3.1 Dataset

The dataset used for our proposed work was obtained from Kaggle Repository [19] consisting of 1 million tweets collected from Twitter randomly. The dataset titled “Sentiment Dataset with 1 Million Tweets” is mainly recognized as a sentiment dataset and has been annotated mainly with labels (positive, negative, uncertainty, litigious). The proposed work mainly deals with positive and negative tweets from the dataset with a total of 526,765 tweets in English language. Statistical analysis of the dataset is shown in Table 2.

Table 2. Statistical Data from Dataset

Description	Positive Tweets	Negative Tweets
Number of Tweets	262220	264545
Mean Tweet length (in words)	23.52	26.06

3.2 Data Preprocessing

Since most of the tweets contained in the dataset have unfiltered information (urls, hashtags, user mentions and emoji/emoticons), pre-processing the data is a required step. Below are the steps taken to ensure that noise and unwanted data is removed from the raw tweets.

- Removal of links: Using Regular expression, hyperlinks, urls and user mentions can be removed from the raw tweets
- Removal of emoticons: The tweets are processed for removal of emojis, symbols, pictographs, transport and map symbols using their respective unicodes.
- Removal of html related content: Tweets might sometimes contain html embeddings to render image or render some iframe content. All of the html related content are filtered out in this step of process
- Removal of Stop Words: In order to remove stopwords (“the ", "a ","an ") from the tweets, python library “nltk” is used. Its dictionary of English stop words are used to remove all the stop words which don’t contribute to the meaning of the sentence

After all these processes are performed, the raw tweets are converted into cleaned sentences. It is then split into words and all of the individual letters are transformed into lowercase for further processing. The most high frequency words represented in the word cloud is shown in Fig 1 and 2.

Fig. 1. High Frequency Positive words

Fig. 2. High Frequency Negative Words

Lemmatization: The main idea of this technique is to reduce the word into its root form in order to simplify the meaning of the actual sentence. The words from the previous process is lemmatized and is reduced to its root form and it then passed to the tokenizer

Tokenization: In this pre-processing step, vectorizing into a sequence of integers using the text corpus is executed. Lemmatized words are then treated with the tokenization process. The tokenizer is first fitted on the cleaned text and then padded up to a certain max length.

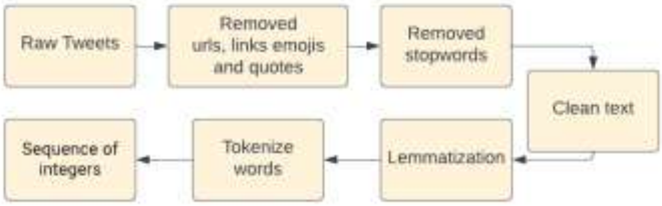

Fig. 3. Flow of preprocessing

Fig 3. shows the overall flow diagram of pre-processing starting from the raw tweets until it undergoes the tokenization process. The processed data is split into ratios 80-10-10 % for training, testing and validation datasets.

3.3 Architecture

Convolution Neural Networks (CNN):

CNN is majorly recognized classified as a category of deep neural networks used in analyzing visual imagery. While traditionally used in computer vision, CNNs [22] are gaining momentum in the domain of Natural Language Processing (NLP), showing promising results. The convolution layer obtains an input sequence of t-words $[t_1, t_2, t_3 \dots . t_n]$ and is represented as X. The equation $X \in R_t * d$ denoting the concatenated word vector, here d represents the dimension of the individual word. A convolution filter $W_f \in R_j * d$ is processed on a window of j words to find out new features represented by $X(i{:}\,i+j-1) \in R_j * d$. The new features obtained after performing convolution operations are denoted by c.

$$c_i = \sigma\big(W.X_{(i:i+j-1)} + b\big) \,\epsilon\, R \tag{1}$$

In the equation (1), σ represents the convolution operator, it further denotes the concatenation of word embeddings from i^{th} word to $(i+j-1)^{th}$ word. b will be the bias for W and W is the weight vector. σ denotes the rectified Linear unit (ReLU) activation function. In order to achieve the map of features, convolution operation is applied on to the whole text represented by X.

$$c = \big[c_1, c_2 \dots c_i \dots . c_{n-j+1}\big] \,\epsilon\, R_{n+j+1} \tag{2}$$

Max-pooling layer is then utilized in obtaining the significant features from the feature map. After the output is achieved from the convolution layer, it then passed into the Softmax-activation function to predict on the final output of the model.

The approach of using Convolutional Neural Networks (CNNs) for Twitter sentiment analysis has its limitations and potential shortcomings. CNN algorithms frequently miss the sequential context of the data, but they are more effective at identifying local patterns in text categorization. When it comes to sentiment analysis, Transformer-based models like Recurrent Neural Networks (RNNs) do better at collecting contextual information and longer-range relationships that are necessary to grasp the meaning of a tweet. CNNs often have trouble constructing good generalisations from Twitter data since it might be noisy and sparse. To function well, CNNs require a large volume of training data.

Recurrent Neural Networks (RNN):

An RNN is a neural network designed for processing sequential data or time series data. Unlike feedforward networks, RNNs possess a limited memory capacity, which means that the current output at a specific time step is dependent on the input given by the previous time step. Making RNNs an intricate tool for handling ordered data or data where sequence is significant.

In the proposed model, LSTM and Bi-directional LSTM are utilized to implement RNN in the architecture. The long-term dependency problem in RNNs is solved through the use of LSTM. LSTM utilizes a memory cell to store information over a long period of time, and mainly three gates (input, output, and forget) to regulate the flow of information into and out of the cell.

Equations relating to input gate, forget gate, and output gate are as follows:

$$i_t = \sigma(W_i[h_{t-1}.x_t] + b_i) \tag{3}$$

$$f_t = \sigma\big(W_f[h_{t-1}.x_t] + b_f\big) \tag{4}$$

$$O_t = \sigma(W_o[h_{t-1}.x_t] + b_o) \tag{5}$$

i_t→ denotes the input gate.
f_t→ denotes the forget gate.
O_t⟶ denotes output gate.

$\sigma \rightarrow$ represents sigmoid function.
$W_x \rightarrow$ denotes the respective gate(x) weight.
$h_{t-1} \rightarrow$ denotes the previous LSTM block (at timestamp t - 1) output.
$x_t \rightarrow$ denotes the current timestamp input.
$b_x \rightarrow$ denotes biases for the each gates(x).

The equation for the control gate is as follows:

$$g_t = \tanh(W_g * [h_{t-1}.x_t]) \tag{6}$$

where g_t is the value of the control gate.
Equation for the current memory cell state is shown below:

$$c_t = f_t * c_{t-1} + i_t * g_i \tag{7}$$

where ◌ is the value of the current memory cell state.

Finally, the equation for the output of the LSTM block is as follows:

$$h_t = o_t * \tanh(c_t) \tag{8}$$

where h_t is the LSTM block output

These equations are used to calculate the values of the input gate, forget gate, output gate, control gate, memory cell state, and output of the LSTM block at each time step. The LSTM architecture is trained using backpropagation through time to minimize the loss function and improve the model's accuracy.

BiLSTM [26] effectively addresses the limitations of traditional unidirectional LSTMs by processing sequential data in both forward and backward directions simultaneously. Bidirectional LSTM design consists of two unidirectional LSTMs that process the sequence in both forward and backward directions. This design can be viewed as having two independent LSTM networks, one receiving the token sequence as it is and the other receiving it in reverse order. This enables a more comprehensive understanding of context, the modelling of long-range dependencies, and improved feature extraction, making BiLSTM's particularly suitable for tasks where word order.

LSTM networks have a more complex architecture compared to CNNs, with more gates and parameters. This complexity can make them computationally expensive and more challenging to train and tune. Due to their complex architecture, LSTMs are often prone to overfitting which causes the LSTM model to perform exceptionally well on the training data but significantly worse on new information, where its performance is not as robust due to its tendency to capture noise and peculiarities from the training dataset. Training LSTMs is inherently sequential and less suited for parallel processing. This can result in slower training and inference times compared to highly parallelized CNNs. Fig.4 shows our proposed Model Architecture with CNN and Bidirectional LSTM layers

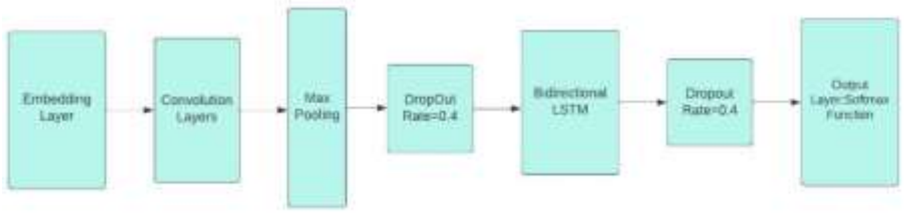

Fig. 4. CNN + BiLSTM Model Architecture

The proposed model is a hybrid of Bidirectional LSTMs and CNNs, as shown in the Fig.4. This model aims to mitigate the problems of both CNNs and LSTMs, resulting in a more robust model that can accurately predict the sentiment of tweets.

Our proposed model is composed of six layers: an embedding layer, a CNN layer, a max pooling layer, a Dropout layer, a Bidirectional layer, and a dense layer. These layers work cohesively to predict the final output of the deep learning architecture. The embedding layer takes an input sentence converts into its embeddings and then passes it to the CNN layers. The CNN part of the architecture extracts the high-level features from the sequence of words $[w_1, w_2, w_3 \dots . w_n]$ associated with embedding vectors of dimension d which were received from the embedding layer. Then the output is sent to the max pooling layer.

Max Pooling layer in this hybrid architecture helps by down sampling the feature maps generated by the CNN, which reduces computational complexity and retains the most crucial local features. This processed data is then fed into the LSTM part of the architecture. This LSTM part of the model consists of Bidirectional LSTM layers in which both past and future contexts have been modelled by considering temporal information flow in both directions that is forward and backward. This helps the model to understand the entire context of the tweet and make accurate predictions. Finally, the architecture also includes some dropout layers [21] which helps to overcome the overfitting problem caused due to the complexity of the model.

4 Experimental Results

In this step, our proposed work evaluates results from three different deep learning models i.e. CNN, LSTM and hybrid of CNN and Bidirectional LSM. The results were compiled using Google Collab's GPU. For the training purpose, the training data consist of 80%, testing data 10 % and validation data 10 % of the original dataset. Adam Optimizer is used to compile the model over 20 epochs. We have observed that the ensemble model (CNN+Bidirectional LSTM) model has achieved an accuracy of 98.16 %. But other two deep learning models are quite close to the accuracy model with CNN with an accuracy of 92.42 % and Bidirectional LSTM model with an accuracy of 91.26

%. The performance metrics used to evaluate the different models are accuracy, precision, f1-score and recall as shown in Table 3.

Table 3. Classification report for mentioned models

Deep Learning Model	Accuracy	Precision	Recall	F1-score
CNN	92.42	92.31	92.14	92.63
Bidirectional LSTM	91.26	91.45	91.38	91.63
CNN +Bidirectional LSTM	98.16	98.16	98.16	98.16

The accuracy during training of the model over the number of epochs is shown in Fig.5 and Fig.6 shows the loss curve over epochs. We can observe that the difference between the training and validation accuracy curves is not too large, indicating that there is some but not major over fitting.

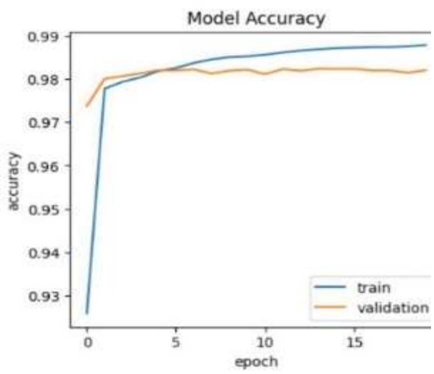

Fig. 5. Accuracy of training and validation set over epoch

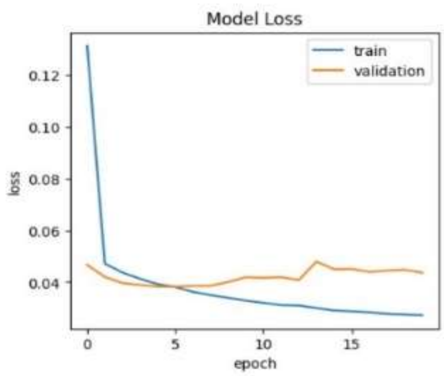

Fig. 6. Loss on training and validation set over epoch

5 Conclusion and Future Work

In today's day and age, understanding and analyzing the user's sentiment is crucial for determining the trajectory of businesses, organization and policy making. Our proposed work highlights the significance of using a hybrid deep learning model to evaluate user's perception. The CNN and Bidirectional LSTM model having an accuracy of 98% has proved to capture the local and the global contextual information. With the model's high accuracy, it can be highly effective in monitoring content on social media and determining user behavior from his current activities

Future works can include having a customized dataset for training which solely focuses on vocabulary relating to positive and negative sentiment words which results in better classifying the data into categories. Multilingual tweets can be trained to predict sentiments in different languages. As an extra step in pre-processing, emoji's can be expressed to correlate their literal meaning. The same process can be applied to slang language and sarcastic language.

Abbreviations	Definitions
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
DL	Deep learning
SVM	Support Vector Machine
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory

References

1. Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment Analysis of Tweets using SVM. International Journal of Computer Applications, 177. http://dx.doi.org/10.5120/ijca2017915758
2. Ahmad, M., Aftab, S., Muhammad, S. S., & Ahmad, S. (2017). Machine Learning Techniques for Sentiment Analysis: A Review. International Journal of Multidisciplinary Sciences and Engineering, 8(3).
3. Alamoudi, E. S., & Alghamdi, N. S. (2021). Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. Journal of Decision Systems, 30, 259-281. https://doi.org/10.1080/12460125.2020.1864106
4. Antonakaki, D., Fragopoulou, P., & Ioannidis, S. (2020). A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks. Expert Systems with Applications, 164. https://doi.org/10.1016/j.eswa.2020.114006
5. Bello, A., Ng, S.-C., & Leung, M.-F. (2022). A BERT Framework to Sentiment Analysis of Tweets. Sensors. https://doi.org/10.3390/s23010506
6. D, G., & S, K. (2019). Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data,. 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 12-17. https://doi.org/10.1109/Deep-ML.2019.00011
7. Drus, Z., & Haliyana, K. (2019). Sentiment Analysis in Social Media and Its Application: Systematic Literature Review. Procedia Computer Science, 161, 707-714. https://doi.org/10.1016/j.procs.2019.11.174
8. E, P., M, S., Hari Krishnan, A., Kumar, T., & Sudhakar, R. (2019). Sentiment Analysis of US Airline Twitter Data using New Adaboost Approach. International Journal of Engineering Research & Technology.
9. Gupta, B., Negi, M., Vishwakarma, K., Rawat, G., & Badhani, P. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. International Journal of Computer Applications, 165. http://doi.org/10.5120/ijca2017914022
10. Jain, P., Pamula, R., & Srivastava, G. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. Computer Science Review, 41. https://doi.org/10.1016/j.cosrev.2021.100413

11. Jindal, S., Tyagi, A. K., & Sharma, K. (2020). Twitter user's Behavior and Events Affecting Their Mood. International Journal of Engineering and Advanced Technology (IJEAT), 9(3). https://doi.org/10.35940/ijeat.C6121.029320
12. Karami, A., Lundy, M., Weeb, F., & Triwedi, Y. (2020). Twitter and Research: A Systematic Literature Review Through Text Mining. IEEE Access, 8, 67698-67717. https://doi.org/10.1109/ACCESS.2020.2983656
13. Mandloi, L., & Patel, R. (2020). Twitter Sentiments Analysis Using Machine Learning Methods. 2020 International Conference for Emerging Technology (INCET). https://doi.org/10.1109/INCET49848.2020.9154183
14. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal. https://doi.org/10.1016/j.asej.2014.04.011
15. Mostafa, R., Mehedi, M., Md, A., & Rasel, A. (2022). Bidirectional LSTM and NLP based Sentiment Analysis of Tweets. Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022). SoCPaR 2022. Lecture Notes in Networks and Systems, vol 648. Springer, Cham. https://doi.org/10.1007/978-3-031-27524-1_62
16. 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 by MDPI, 21. https://doi.org/10.3390/s21072266
17. Osatuyi, B. (2013). Information sharing on social media sites. Computers in Human Behavior, 29(6), 2622-2631. https://doi.org/10.1016/j.chb.2013.07.001
18. Qian, Y., Deng, X., Ye, Q., Ma, B., & Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. Information Processing & Management, 56(6). https://doi.org/10.1016/j.ipm.2019.102086
19. Sentiment Dataset with 1 Million Tweets. (n.d.). Retrieved October 15, 2023, from Kaggle: https://www.kaggle.com/datasets/tariqsays/sentiment-dataset-with-1-million-tweets
20. Severyn, A., & Moschitti, A. (2015). Twitter Sentiment Analysis with Deep Convolutional Neural Networks. SIGIR '15: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. https://doi.org/10.1145/2766462.2767830
21. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research, 1929-1958.
22. Yadav, N., Kudale, O., Rao, A., Gupta, S., & Shitole, A. (2020). Twitter Sentiment Analysis using Supervised Machine Learning. Intelligent Data Communication Technologies and Internet of Things, 57, 631–642. https://doi.org/10.1007/978-981-15-9509-7_51
23. Zeglen, E., & Rosendale, J. (2018). Increasing Online Information Retention: Analysing the Effects of Visual Hints and Feedback in Educational Games. Journal of Open, Flexible and Distance Learning. https://doi.org/10.61468/jofdl.v22i1.325
24. Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation Classification via Convolutional Deep Neural Network. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2335–2344
25. Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. https://doi.org/10.48550/arXiv.1801.07883
26. Zhang, S., Zheng, D., Hu, X., & Yang, M. (2015). Bidirectional Long Short-Term Memory Networks for Relation Classification. 29th Pacific Asia Conference on Language, Information and Computation, 73 - 78