INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

**Received:** 28-10-2023

**Accepted:** 13-12-2023

**Published:** 12-01-2024

**Citation:** Prabhu R, Nashappa CS (2024) An Aspect-Based Sentiment Analysis Model to Classify the Sentiment of Twitter Data using Long-Short Term Memory Classifier. Indian Journal of Science and Technology 17(2): 184-193. https://doi.org/10.17485/IJST/v17i2.2715

* **Corresponding author**.

rakshitha79@gmail.com

**Funding:** None

**Competing Interests:** None



Published By Indian Society for Education and Environment (iSee)

**ISSN**
Print: 0974-6846
Electronic: 0974-5645

# An Aspect-Based Sentiment Analysis Model to Classify the Sentiment of Twitter Data using Long-Short Term Memory Classifier

**Rakshitha Prabhu[1]*, Chandrashekara Seesandra Nashappa[2]**

**1** Department of Computer Science and Engineering, S.J.C. Institute of Technology, Chikkaballapur, Karnataka, India
**2** Department of Computer Science and Engineering, C. Byre Gowda Institute of Technology, Kolar, Karnataka, India

## Abstract

**Objectives**: To design an effective sentiment analysis model for interpretation of polarities in the text, that remains a challenging task while classifying the sentiment of tweets. **Methods**: This research proposes Aspect-Based Sentiment Analysis (ABSA) model which effectively retrieves the original contextual meaning of the text and helps in effective text categorization. The feature extraction is performed using Bag of Words (BOW) and Time-Frequency – Inverse Document Frequency (TF-IDF). The extracted features undergo the process of aspect-based sentiment analysis, which mines out the set of aspects to determine polarity of the text. Finally, the contextual words are classified using Long-Short Term Memory (LSTM), to determine the polarity as positive, neutral and negative. **Findings**: The experimental results show that the proposed ABSA with LSTM offers better results in classification accuracy with 93.54% which is relatively higher than the existing techniques by name of Stochastic Gradient Descent optimization based on Stochastic Gate Neural Network (SGD-SGNN) and Binary Brain Storm Optimization and Fuzzy Cognitive Maps (BBSO-FCM) which attained the accuracies at 90.67% and 88.71% respectively. **Novelty**: The ABSA is implemented to extract the set of aspects and determine polarities of the tweet obtained from extracted features. In aspect-based extraction, every individual token is labeled with contextual embedding, Part-of-Speech (PoS) embedding and embedding based on dependencies. The syntactic presentations at the finalized stages are improvised by combining the hidden features, appeared PoS states and the states based on dependencies. Thus, the proposed approach helps to minimize the complexity while classifying the sentiments using LSTM.

**Keywords:** Tweets; Twitter Data; Aspect-based sentiment analysis; Bagof word; Long-short term memory

## 1 Introduction

Recently nowadays, the usage of microblogging social platforms has reached its peak since the social media platforms such as Twitter, Instagram, Facebook, etc. generate a connection among users to instantly share their thoughts and opinions [(1)]. Among these, Twitter is considered an active platform with more users and helps business mediums to improvise their standards by collecting data on how people remark on their services and products [(2,3)]. Moreover, Twitter acts as a platform to express the ideas and opinions of the public through tweets. Sentiment analysis is one of the research areas in Natural Language Processing (NLP) which helps in the detection of sentiment present in the tweets. Many people use the SA as a tool to evaluate reviews, polls, economic reports, etc. [(4,5)]. During the time of sentiment analysis, tokenization is performed to tokenize the words and phrases present in the tweets. Twitter uses a global language that lacks correct grammar and different slang. Therefore, to improve the SA findings, a pre-processing step must be included to eliminate the undesired data. The sentiment analysis is performed using both machine learning and deep learning techniques. Among these, deep learning techniques provide more reliable outcomes during classification of text.

The ability of deep learning techniques in modelling non-linear and improper data structures into an organized form of data, helps it to attain better results than the machine learning algorithm [(6)]. Generally, tweets are gathered from the Twitter API database and then pre-processing techniques are employed to remove the stop words, blank spaces, punctuation, and other useless information [(7)]. Feature extraction is an essential process to extract features from the pre-processed tweets. After the process of feature extraction, the pre-processed tweets are transformed into features [(8)]. The relevant features from the extracted features are chosen using feature selection and this is a vital step that simplifies the process of text classification. The Tweets are categorized based on their polarities that are, positive, negative and neutral [(9,10)]. The existing researches based on sentiment analysis of Twitter data had drawbacks such as constrained learning rate, dissimilar labels, and incapability in analyzing sentiment of the word aspect. To overcome the aforementioned problems, this research proposed a sentiment analysis model using aspect-based analysis and LSTM as a classifier. The ABSA model introduced in this research effectively labels every individual token with contextual embedding, PoS embedding and embedding based on dependencies. Moreover, the syntactic presentations in the finalized stages are improvised by combining the hidden features, appeared PoS states and the states based on dependencies. Thus, the ABSA helps in the process of mining the aspects, which minimize the complexity while categorizing the sentiment of tweets. The recent related works based on sentiment analysis of Twitter data, using machine learning and deep learning techniques, are as follows:

AlBadani et al. [(5)] introduced an effective method to analyze the sentiment of Twitter data using deep learning architectures such as Universal Language Model Fine Tuning (ULMFiT) along with Support Vector Machine (SVM). The introduced method utilized Back-Propagation Through Time (BPTT) to categorize the text and model the language. The introduced methods identified the people's attitudes toward products based on their comments. However, the ULMFiT method with SVM was incapable of evaluating the sentiment at aspect levels.

Gandhi et al. [(6)] developed a sentiment analysis framework using Convolutional Neural Network (CNN) and LSTM to evaluate the tweets based on their polarities. Initially, the reviews were tokenized into token words and the tokenized words were passed through the embedding layer which transformed the word into a predefined size. Then, the classification takes place using the CNN-LSTM which categorizes the sentiment of the tweets effectively. The introduced method overcame the issues regarding user reliability.

Kothamasu and Kannan [(7)] introduced a deep neural network by combining the Improved Adaptive-Network based Fuzzy Inference System (IANFIS) to analyze the sentiment of people on online products. The tweets of the people were collected from Twitter data and the feature extraction was performed using the Spider Monkey Optimization (SMO) algorithm. Finally, the extracted features were passed through the layers of IANFIS to categorize the best products based on the public's opinion. However, the SMO utilized in the proposed method was incapable of solving multi-objective issues.

Bibi et al. [(8)] introduced a framework using unsupervised learning which was a concept-based hierarchical clustering, to analyze the sentiment of Twitter data. The framework utilized ensemble-based hierarchical clustering and concept-based method to extract Bag of Concepts (BOC). Moreover, the proposed method utilized the polarity intensity criteria to detect polarity of the sentences. The Bayesian model utilized in the framework was capable of categorizing the delegated tweets present in the Twitter data. However, the concept-based approach utilized in the framework was not up to the mark due to its constrained knowledge base.

Vidyashree and Rajendra [(9)] introduced an improved sentiment analysis model by utilizing Stochastic Gradient Descent (SGD) optimization algorithm based on Stochastic Gate Neural Network (SGNN) to detect the polarities of the tweets. The features were extracted from the pre-processed output and the dimensionalities of the features were minimized using linear discriminative analysis. After this, weighted feature selection took place and the tweets were classified using SGNN. The SGD algorithm utilized in this research contains more labels and it was not limited to time constraints. However, the introduced method did not suit well for word-level classification.

Jain et al.[10] developed a Binary Brain Storm Optimization and Fuzzy Cognitive Maps (BBSO-FCM) to analyze the sentiment of tweets. The developed model underwent pre-processing to eliminate the undesired words. After this, TF-IDF was used to extract the features and the proposed BBSO algorithm was used to select the relevant features. The improved classification was accomplished using Fuzzy Cognitive Maps (FCM) which categorized the tweets as positive and negative based on polarities. However, the softmax utilized in this research seemed to be dissimilar while predicting the labels.

### 1.1 Research Gap

Overall, the existing researches faced problems related to classifying sentiments at the aspect levels. Moreover, the dissimilarity occurred while predicting the labels of individual tokens due to constrained knowledge. Some, of the existing researches did not analyze the sentiments with proper efficiency, at word level, due to improper extraction of features. To overcome these limitations of the existing approaches, an aspect level sentiment analysis model is developed which effectively analyses the sentiments and aids in better classification results.

The major contribution of this research is listed as follows:

(1) Aspect-Based Sentiment Analysis (ABSA) model is proposed which effectively retrieves the original contextual meaning of the text and helps in effective text categorization. The ABSA effectively labels the individual token and combines the hidden features which helps to finalize the aspects.

(2) The features from the pre-processed output is extracted using BOW and TF-IDF techniques. The features are analyzed using ABSA, and then classified based on their polarities using LSTM classifier. The complexity of the LSTM classifier is diminished due to usage of ABSA which finalizes the syntactic presentations and dependencies.

This research paper is structured as follows: The related work is provided is in the latter part of section 1 and the methodology of the proposed technique is shown in section 2. The results and analysis are provided in section 3 and finally, the conclusion of this research is presented in section 4.

## 2 Methodology

### 2.1 Sentiment Analysis of Tweets using Aspect-Based Feature Extraction

This research introduces an aspect-based sentiment analysis model to categorize the sentiment of tweets using LSTM classifier. The main objective of this research is to classify the tweets at the aspect level, using LSTM classifier. In the proposed research, the sentiment of the tweets is extracted using aspect-based concept, which analyses the aspect of the word and classifies it based on the polarity. Initially, the Twitter data is collected from Twitter API dataset, in the form of raw data, which is then pre-processed using tokenization and lemmatization, stop word removal and Word2Vec. The features from the pre-processed output are then extracted using BOW and TF-IDF. The extracted features undergo aspect-based analysis to identify the aspect of every individual word in the sentence and the classification is performed using LSTM classifier. The overall process involved in classifying the sentiment of tweets is presented in Figure 1 as follows:

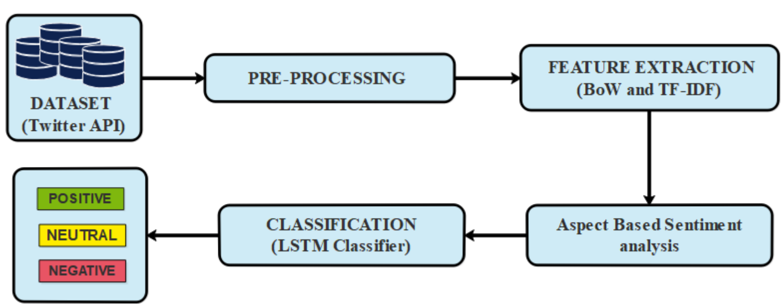


**Fig 1. The overall process involved in sentiment analysis**

### 2.2 Data Acquisition from Twitter API

This research utilized the Twitter API dataset to collect sample tweets of users, lists, comments etc.,[11]. Moreover, it is an open-source dataset regarding different types of topics that can be accessed by the public. The tweets contained in Twitter API are in a raw form (i.e. with a long sentence, misspelled words and stop words). These redundant things may affect the classification accuracy so a pre-processing method is essential to remove the unwanted data.

### 2.3 Pre-processing

The raw data comprises unnecessary data with irrelevant texts, and numerous redundancies which may affect the overall performance. To remove this, the pre-processing technique is essential in reducing the complexities of the overall performance. Moreover, the raw data obtained from the Twitter dataset is not suitable for direct analysis as it comprises many irrelevant data. In data pre-processing, the raw data is converted into a readable format which helps to enhance the overall accuracy. In pre-processing, the quality of the raw data is improved by the following methods.

(i) Tokenization and lemmatization[12]: In tokenization, the text is fragmented into smaller texts known as tokens. Every token obtained from the tweets is stored in a token list and these tokens appear based on their original order. In lemmatization, the words obtained from the tweets are converted into dictionary form.

(ii) Stop word removal[12] – The stop words are a collection of NLP techniques that neglect the irrelevant information from the sentence. Repeated words like adverbs, prepositions, and articles are removed from the tweets. The removal of this irrelevant data helps to lower the dimensionality of the data which helps in improved classification performance.

(iii) Word2Vec – It is an embedding model which maps the tweets into their vector presentation wherein each word present in the tweet is converted to its corresponding numerical vector. This vector format is utilized to combine the dimensionalities of LSTM with the output array.

### 2.4 Feature extraction using Bag of Words and TF-IDF

The BOW model is used in the process of feature extraction where the appropriate features are filtered from the pre-processing output. The BOW is also defined as the group of words which is utilized to label the contextual sentence along with the count of words. There are two phases involved in the BOW model, listing the well-known terms and the second one is to regulate the existence of well-known terms. Initially, the vocabularies of each unknown word present in the tweets are identified and a list of these distinct words is created to observe their presence in each tweet. At last, the matrix of numbers is passed through the BOW model to train it to its fullest. This aids in labeling the contextual words in the sentence.

After processing the text with BOW, the TF-IDF[13] method is utilized to evaluate the significance of the word in a tweet. Moreover, it describes the degree of contextual similarity of the words and detects the similarities among the words. The significant texts present in the document are evaluated using IDF that particulate the significant text in the tweets. Moreover, IDF is used to evaluate rarely used words present in the document. The value of TF is computed using Equation (1) presented below,

$$TF(t,d) = \frac{N_{(t,d)}}{T} \tag{1}$$

Where the term frequency of the terms present in the document is considered as $TF(t,d)$, the occurrence of the term in the document is denoted as $N_{(t,d)}$ and total number of terms is represented as $T$. Thus, for different tweets, different $TF(t,d)$ values are evaluated. Equation (2) provides the formula to evaluate IDF value.

$$IDF(t) = logN/N(t) \tag{2}$$

Where the inverse document frequency of the term $t$ is denoted as IDF and the number of text is denoted as $N(t)$. From the aforementioned Equations (1) and (2) about TF and IDF, the value of TF-IDF is evaluated using Equation (3) as follows:

$$TF - IDF = TF \times IDF \tag{3}$$

After the stage of extracting features from TF-IDF, the aspect-based analysis is performed which is explained in the following sections.

### 2.5 Aspect-based sentiment analysis

The tweets obtained from extracted the output undergo the stage of aspect-based analysis wherein the aspect of every individual feature is selected using the proposed word Aspect Based Sentiment Analysis (ABSA) model. The aspect-based analysis is performed to extract the set of aspects and determine the polarities of the tweet obtained from extracted features. In aspect-based extraction, every individual token is labeled with contextual embedding, Part-of-Speech (PoS) embedding and embedding based on dependencies. The syntactic presentations at finalized stages are improvised by combining the hidden features, appeared PoS states and the states based on dependencies. The process involved in the proposed aspect-based feature extraction is represented in Figure 2 as follows:

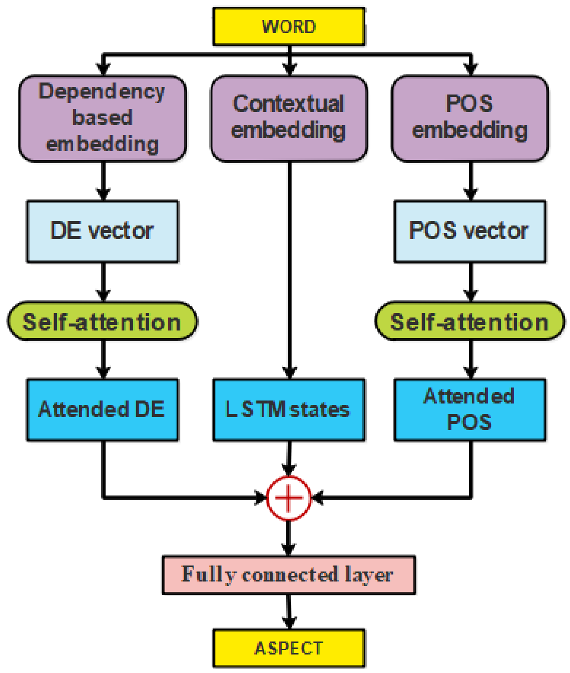


**Fig 2. The process involved in aspect- based analysis**

a) Demonstration of Input: The aspect-based model requires distinctive categorization tokens referred to as $T$, input sequence at the beginning, and separator $S$ at the end of the sequence. Thus, the format of the sentence should be changed before processing it into the aspect-based model. The overall format of the input sentence is converted based on the form of $T+$ input sequence $+S$. b) PoS embedding: The PoS of every individual word is interpreted using PoS tagging and subsequently, PoS of the input sequence is retrieved. The input sequence of the PoS tagging is represented in Equation (4) as follows:

$$P = \{p_1, p_2, \ldots, p_n\} \tag{4}$$

The embedded layer of PoS obtains a sparse representation of vectors to gain the dense vector representation, which is presented in Equation (5) as follows:

$$V^p = \left\{v_i^P \mid i \in [1, n]\right\} \tag{5}$$

Where $v_i^P \in R^{hPOS}$ and the $hPOS$ refers to the hidden side of POS embedding.

The process of POS embedding utilizes the whole sequence of POS taggers and extracts grammatical dependencies from the input sequence.

c) Embedding based on dependencies: The dependency-based embedding utilizes dependent contexts based on syntax relation among the participant words. Every individual target word $W$ consists of modifiers attached to it. The obtained context $C$ is represented in Equation (6) as follows:

$$C = \{(M_1, Rel_1), (M_2, Rel_2), \ldots, (M_n, Rel_n)\} \tag{6}$$

Where $Rel$ is known as the relation of dependencies between the target word $W$ and the modifiers $M$.

During the prior period of final context extraction, the contexts containing prepositions are combined by incorporating the prepositions of the labeled dependencies. The dependency-based embedding integrates the withdrawn connection of linear word embedding. Moreover, the inappropriate texts are de-highlighted, which tend to fall in contextual windows.

a) Fine-tuning: Finally, the final contextual words obtained from aspect-based feature extraction are tuned to minimalize the loss rate of cross entropy. The fine-tuning takes place using $L_2$ regularization norm, using optimistic parameters. The process of fine-tuning is performed based on the Equation (7) as follows:

$$L(\theta) = -\sum_{i=}^{n} \widehat{y}_i log y_i + \lambda \sum_{\theta \in \Theta} \theta^2 \tag{7}$$

Where the regularization and optimistic parameters are denoted as $\lambda$ and $\theta$. The expected textual labels that correspond to $y_i$ are denoted as $\widehat{y}_i$.

## 2.6 Classification using LSTM

The aspect-based analysis provides fine-tuned contextual texts which are needed to be classified based on the polarity of the sentence. This research utilizes LSTM[14,15] as a classifier to categorize the text as positive, neutral, or negative, based on polarities. LSTM is one of the types of recurrent neural networks which retains information related to long-term dependencies. The architectural diagram of the LSTM is shown in Figure 3 as follows:

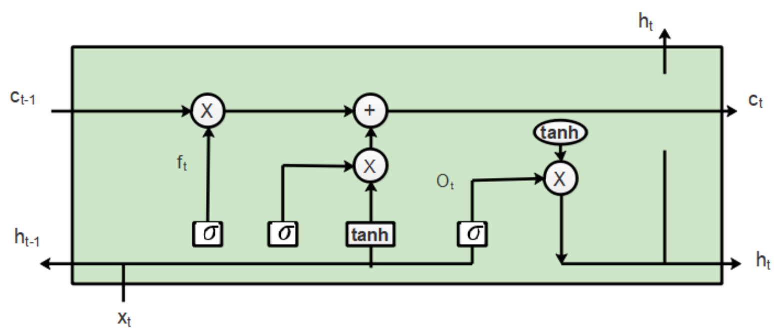


**Fig 3. Architectural diagram of LSTM**

The LSTM typically contains a current cell state $(C_t)$ and three gates that are, input gate $(i_t)$, output gate $(O_t)$, and forget the gate $(f_t)$. The equation of the cell state of LSTM is represented in Equation (8) as follows:

$$C_t = C_{t-1} \times f_t + C_t \times i_t \tag{8}$$

The gradient present in the LSTM is in a preserved form that eradicates the issues based on vanishing the gradients. So, LSTM is considered an effective method for word sequence and modeling languages. Each node present in LSTM receives $C_{t-1}$ from the prior nodes of LSTM and obtains a word vector $x_t$ and the output vector as $h_{t-1}$. The forget gate is utilized to forget and keep the information in the cell state and the mathematical form of forget gate is represented in Equation (9) as follows:

$$f_t = \sigma \left( W_f . [h_{t-1}, x_t] + b_f \right) \tag{9}$$

Where the sigmoid function is denoted as $\sigma$ that provides an output lying in the range of 0 to 1. The input layer is responsible for deciding whether the value of the next input is to be considered by the preceding layer or not. The current state with updated vector value is represented as $\widetilde{C_t}$ which is represented in Equation (10) and the Equation (11) defines the equation of the input gate.

$$\widetilde{C_t} = tanh \left( W_C . [h_{t-1}, x_t] + b_C \right) \tag{10}$$

$$i_t = \sigma \left( W_i . [h_{t-1}, x_t] + b_i \right) \tag{11}$$

Where the weighted state and the biased state are denoted as $W$ and $b$ respectively. The output $O_t$ of the LSTM and $h_t$ is represented in Equations (12) and (13) respectively.

$$O_t = \sigma\left(W_0.\left[h_{t-1}, x_t\right] + b_0\right) \tag{12}$$

$$h_t = O_t \times tanh\left(C_t\right) \tag{13}$$

The data will be transferred from one layer to another layer in sequential order and to every hidden layer, using the *tanh* activation function. The output layer utilizes the softmax activation function that produces an output that lies within the range of [0,1] which identifies and categorizes the sentiment of the particular tweet.

## 3 Results and Discussion

This section provides a detailed analysis of results obtained by evaluating the efficiency of the proposed method, on the basis of the performance metrics, accuracy, precision, recall and F1 score. Moreover, this section is segregated into two sub-sections named as performance analysis and comparative analysis. The performance of the classifiers is evaluated in the presence and absence of the ABSA model, while in the comparative analysis the efficiency of the LSTM-ABSA is evaluated and compared alongside the existing classification techniques, SGD-SGNN[9] and BBSO-FCM[10].

### 3.1 Experimental setup

In this research, the performance of ABSA is evaluated using Python 3.7 software with system specifications such as an Intel i7 processor at 2.20 GHz and 8 GB RAM capacity in the Windows 11 operating system. The Twitter API dataset is utilized to calculate the efficacy of the proposed ABSA model.

### 3.2 Performance metrics

The performance of the proposed ABSA is evaluated based on accuracy, precision, recall and F1 score, which are represented in Equations (14), (15), (16) and (17). The description of the aforementioned metrics is described in this section.

**Accuracy**

It is defined as the total number of truly classified text to the total number of text present in the document.

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \tag{14}$$

**Precision**

It is defined as truly positive text to the total number of positive texts.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

**Recall**

It is stated as truly positive text to false negative and true positive texts.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

**F1-score**

It is evaluated based on considering the value of both recall and precision

$$F1\text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

### 3.3 Performance analysis

In this section, the resultant performance of the proposed ABSA is evaluated with Context Based Sentiment Analysis (CBSA) model and Rule Based Sentiment Analysis (RBSA) as per accuracy, precision, recall, and F1 score metrics. Table 1 presents the performance of the proposed ABSA, along with those of the existing CBSA and RBSA. The results from Table 1 show that the proposed ABSA, on an overall basis, achieves better results in all the evaluation metrics. The proposed ABSA achieved better classification accuracy of 93.54% which is comparatively higher than the accuracies attained by CBSA and RBSA.

Better implementation results are delivered by the proposed ABSA due to its efficiency in performing analysis at the aspect level, and the labeling of every token in an individual manner. This effectively determines the polarities with higher classification accuracy.

**Table 1. Performance evaluation based on sentiment analysis**

| Models | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|---|
| CBSA | 89.12 | 83.67 | 80.19 | 81.38 |
| RBSA | 90.37 | 87.67 | 85.23 | 83.91 |
| ABSA | 93.54 | 89.71 | 87.28 | 85.21 |

Secondly, the performance of the LSTM classifier is evaluated with existing classifiers, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Generative Adversarial Networks (GAN). The evaluation is performed without aspect-based sentiment analysis, as presented in Table 2.

The results from Table 2 show that the LSTM classifier has achieved better results when compared with other classification models. For instance, the LSTM has achieved a precision of 89.71% which is higher than the precisions of CNN, RNN and GAN measured respectively as 83.22%, 79.31% and 85.44%. The better result of LSTM is due to its capability in handling long-term dependencies and its ability to remember the data over a long period. Moreover, it is less vulnerable to vanishing gradient problems.

**Table 2. Performance evaluation of classifiers without ABSA**

| Classifiers | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|---|
| CNN | 85.61 | 83.22 | 80.12 | 78.43 |
| RNN | 82.89 | 79.31 | 77.66 | 77.57 |
| GAN | 89.64 | 85.44 | 83.11 | 82.12 |
| LSTM | 93.54 | 89.71 | 87.28 | 85.21 |

Similarly, after implementing along with aspect-based feature selection, the performances of the classifiers are evaluated based on accuracy, precision, recall and F1 score. Table 3 shows the results obtained from the classifiers in the presence of the ABSA model.

The results from Table 3 show that LSTM in the presence of ABSA has achieved better classification results when compared with existing classifiers. For instance, LSTM with ABSA has achieved a better classification accuracy of 95.43% which is relatively higher than other classification models. This better classification is due to the proposed ABSA model which employs sentiment analysis for every individual token in the provided data and predicts the sentiment accurately.

**Table 3. Performance evaluation of classifiers with ABSA**

| Classifiers | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|---|
| CNN | 89.17 | 87.24 | 84.78 | 82.34 |
| RNN | 86.89 | 85.12 | 81.39 | 86.12 |
| GAN | 91.47 | 89.90 | 86.77 | 87.54 |
| LSTM | 95.43 | 91.87 | 92.76 | 88.23 |

### 3.4 Comparative analysis

In this section, the performance of the proposed ABSA-LSTM model is compared with existing sentiment analysis models of SGD-SGNN[9] and BBSO-FCM[10]. The results obtained from the comparative analysis are presented in Table 4 as follows,

**Table 4. Comparative table**

| Methods | Accuracy (%) | Precision (%) | Recall (%) | F1 score (%) |
|---|---|---|---|---|
| SGD-SGNN [9] | 90.67 | 85.92 | 83.19 | 80.22 |
| BBSO-FCM [10] | 88.71 | 86.54 | 81.88 | 82.51 |
| ABSA-LSTM | 95.43 | 91.87 | 92.76 | 88.23 |

The results from Table 4 show that the proposed ABSA-LSTM method obtained better performance in the overall evaluation metrics. Especially, it achieved a classification accuracy of 95.43% which is comparatively higher than the existing methods, SGD-SGNN (90.67%) and BBSO-FCM (88.71%). The better result of the proposed approach is due to the features analyzed by ABSA which helps in easier execution of sentiment classification performed by LSTM. Moreover, ABSA analyzes the sentiment of every aspect and labels every individual token in a unique way which helps to determine the polarity of tweets. Thus, the efficiency of ABSA model helps to classify the sentiment much more appropriately than the existing approaches.

## 4 Conclusion

In this research, aspect-based sentiment analysis is proposed to analyze the sentiment of tweets. Twitter API is used to obtain the input as raw data and this raw data is pre-processed by tokenization, lemmatization, and stop word removal. The major objective of the research is to perform sentiment analysis at the aspect level to detect the sentiment of every contextual word present in the tweets. The pre-processed data is fed into the process of feature extraction where the essential features are extracted using Bow and TF-IDF. The obtained features are processed using the proposed ABSA which mines out the set of aspects and determines the polarity of the text. The ABSA extracts the set of aspects and determines polarities of the tweet obtained from extracted features. In aspect-based extraction, every individual token is labeled with contextual embedding, Part-of-Speech (PoS) embedding and embedding based on dependencies. The syntactic presentations at finalized stages are improvised by combining the hidden features, PoS states and the states based on dependencies. Thus, the process of combining the features helps to enhance the efficiency and minimize the complexity during sentiment classification. Finally, the LSTM is utilized to categorize the text based on their polarity, as positive, negative, or neutral. The results from experiments show that the proposed ASBA with LSTM achieved a better classification accuracy of 93.54% which is comparatively higher than the existing methods such as SGD-SGNN (90.67%) and BBSO-FCM (88.71%). This research helps in providing aspect-based sentiment analysis which focuses on every contextual aspect to determine the polarity of the tweets. Future work will most probably be based on tuning hyper-parameters to enhance accuracy rate of the sentiment classification process.

## References

1) Rodrigues AP, Fernandes R, Shetty A, Lakshmanna K, Shafi RM. Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques. *Computational Intelligence and Neuroscience*. 2022;2022:1–14. Available from: https://doi.org/10.1155/2022/5211949.
2) Qi Y, Shabrina Z. Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. *Social Network Analysis and Mining*. 2023;13(1):1–14. Available from: https://doi.org/10.1007/s13278-023-01030-x.
3) Alsayat A. Improving Sentiment Analysis for Social Media Applications Using an Ensemble Deep Learning Language Model. *Arabian Journal for Science and Engineering*. 2022;47(2):2499–2511. Available from: https://doi.org/10.1007/s13369-021-06227-w.
4) Mendon S, Dutta P, Behl A, Lessmann S. A Hybrid Approach of Machine Learning and Lexicons to Sentiment Analysis: Enhanced Insights from Twitter Data of Natural Disasters. *Information Systems Frontiers*. 2021;23(5):1145–1168. Available from: https://doi.org/10.1007/s10796-021-10107-x.
5) Albadani B, Shi R, Dong J. A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Applied System Innovation*. 2022;5(1):1–16. Available from: https://doi.org/10.3390/asi5010013.
6) Gandhi UD, Kumar PM, Babu GC, Karthick G. Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). *Wireless Personal Communications*. 2021. Available from: https://doi.org/10.1007/s11277-021-08580-3.
7) Kothamasu LA, Kannan E. Sentiment analysis on twitter data based on spider monkey optimization and deep learning for future prediction of the brands. *Concurrency and Computation: Practice and Experience*. 2022;34(21). Available from: https://doi.org/10.1002/cpe.7104.
8) Bibi M, Abbasi WA, Aziz W, Khalil S, Uddin M, Iwendi C, et al. A novel unsupervised ensemble framework using concept-based linguistic methods and machine learning for twitter sentiment analysis. *Pattern Recognition Letters*. 2022;158:80–86. Available from: https://doi.org/10.1016/j.patrec.2022.04.004.
9) Vidyashree KP, Rajendra AB. An Improvised Sentiment Analysis Model on Twitter Data Using Stochastic Gradient Descent (SGD) Optimization Algorithm in Stochastic Gate Neural Network (SGNN). *SN Computer Science*. 2023;4(2):1–11. Available from: https://doi.org/10.1007/s42979-022-01607-x.
10) Jain DK, Boyapati P, Venkatesh J, Prakash M. An Intelligent Cognitive-Inspired Computing with Big Data Analytics Framework for Sentiment Analysis and Classification. *Information Processing & Management*. 2022;59(1):102758. Available from: https://doi.org/10.1016/j.ipm.2021.102758.
11) and AE. Data Collection with Twitter API v2. . Available from: https://www.kaggle.com/code/andrewedward37/data-collection-with-twitter-api-v2.
12) Umer M, Ashraf I, Mehmood A, Kumari S, Ullah S, Choi GS. Sentiment analysis of tweets using a unified convolutional neural network-long short-term memory network model. *Computational Intelligence*. 2021;37(1):409–434. Available from: https://onlinelibrary.wiley.com/doi/epdf/10.1111/coin.12415.

13) Xie G, Liu N, Hu X, Shen Y. Toward Prompt-Enhanced Sentiment Analysis with Mutual Describable Information Between Aspects. *Applied Artificial Intelligence*. 2023;37(1):882–898. Available from: https://doi.org/10.1080/08839514.2023.2186432.
14) Das S, Das D, Kolya AK. Sentiment classification with GST tweet data on LSTM based on polarity-popularity model. *Sādhanā*. 2020;45(1):1–17. Available from: https://www.ias.ac.in/article/fulltext/sadh/045/0140.
15) Trisna KW, Jie HJ. Deep Learning Approach for Aspect-Based Sentiment Classification: A Comparative Review. *Applied Artificial Intelligence*. 2022;36(1):1157–1193. Available from: https://doi.org/10.1080/08839514.2021.2014186.