IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 September 2023, accepted 26 September 2023, date of publication 2 October 2023, date of current version 9 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320738

RESEARCH ARTICLE

Deep-Sentiment: An Effective Deep Sentiment Analysis Using a Decision-Based Recurrent Neural Network (D-RNN)

PUTTA DURGA AND DEEPTHI GODAVARTHI
School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh 522237, India
Corresponding author: Deepthi Godavarthi (deepthi.g@vitap.ac.in)

ABSTRACT Sentiment analysis is a sub-domain in opinion mining that extracts sentiments from the users' opinions from text messages. Opinions from E-commerce websites, blogs, online social media, etc., and these opinions are in the form of text, suggestions, and comments. This paper describes the new sentiment analysis model to predict sentiments effectively that can be used to improve product quality and sales. The proposed approach is an integrated model combining several techniques, such as the pre-trained model BERT-large-cased (BLC) for training the dataset. BLC model contains 24-layer, 1024-hidden, 16-heads, 340M parameters. Optimization algorithms can fine-tune pre-trained models, such as BERT, for sentiment analysis tasks. Fine-tuning involves training the pre-trained model on a specific sentiment analysis task to improve performance. Stochastic Gradient Descent (SGD) is the optimized algorithm that helps to analyze the sentiments effectively from the given datasets. The next step is the combination of pre-processing techniques such as Tokenization, Stop Word Removal, etc. The next step focused on Bag-of-Words (BoW) and word embedding techniques like Word2Vec used to extract the features from the datasets. The deep sentiment analysis (DSA) based classification is designed to classify the sentiments based on aspect and priority model to achieve better results. The proposed model combines Aspect and Priority-based Sentiment analysis with a Decision-based Recurrent Neural Network (D-RNN). The experiments are conducted using Twitter, Restaurant, and Laptop datasets available publicly on Kaggle—the proposed model's performance is analyzed using a confusion matrix. The proposed approach addresses various challenges in analyzing the sentiment analysis. Python programming language with several libraries such as Keras, Pandas, and others extracts the sentiments from given datasets. The comparison between the existing and proposed models shows the effectiveness of the sentiment outputs.

INDEX TERMS Sentiment analysis, BERT-large-cased (BLC), stochastic gradient descent (SGD), bag-of-words (BoW), deep sentiment analysis (DSA).

I. INTRODUCTION

Using machine learning techniques, sentiment analysis (SA), a branch of natural language processing (NLP), finds and obtains relevant information from text data, such as views, emotions, and attitudes [1]. Sentiment analysis aims to categorize a text's polarity or whether it reflects a sound, negative, or neutral sentiment. Sentiment analysis uses machine learning methods, including supervised, unsupervised, and deep learning techniques [2]. A machine learning model is trained under supervision using a labelled dataset in which the sentiment polarity of each text sample is tagged [3]. The algorithm then uses this labelled data to discover textual patterns and traits related to sentiment polarity. On the other hand, unsupervised learning techniques do not rely on labelled data but instead identify patterns and clusters in the text data to determine the sentiment polarity. Deep learning approaches involve training neural network models that can

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

learn to extract dynamic features from given input text data without the need for manual feature engineering [4]. Overall, sentiment analysis is a practical application of machine learning with numerous use cases in several domains, including online social networking, market research, client opinion analysis, and more.

Finding the emotional undertone or attitude of a text, whether a social media post, a review, or a news story, is the process of SA. Optimization methods combined with sentiment analysis give improved performance by forecasts and the effectiveness of the analysis process [5]. Gradient descent is one optimization algorithm that can be applied to sentiment analysis. Gradient descent is an algorithm that iteratively modifies a machine learning model's parameters to reduce the discrepancy between anticipated and actual values. This can be used for sentiment analysis by developing an ML model to forecast text sentiment based on elements like word frequency and context and then use gradient descent to modify the model's parameters to increase its precision. Genetic algorithms are an additional optimization technique that can be applied to sentiment analysis [6]. Natural selection, in which a population of potential answers to a problem evolves through time, is the inspiration for the type of optimization method known as genetic algorithms. The performance of a machine learning model can be improved using a set of characteristics or parameters that can be evolved using genetic algorithms in the context of sentiment analysis. Particle Swarm Optimization (PSO) [7], Ant Colony Optimization (ACO), and simulated annealing are optimization approaches used in sentiment analysis [8]. These techniques are also used for feature selection, parameter tuning, and model optimization, among different sentiment analysis-related tasks. Ultimately, by enabling machine learning models to capture the subtleties of human emotions and attitudes more accurately, optimization algorithms can be a potent tool for enhancing the accuracy and efficiency of sentiment analysis [9]. Deep learning techniques are used in many applications, such as predicting pandemic diseases such as COVID-19, chronic kidney Disease (CKD), and other chronic diseases [10]. On the other side, [11] discussed various DL methods that were used to classify Alzheimer's disease (AD) from the real-time datasets using DCNN. A pre-trained model VGG-19 was used to train the model based on abnormalities obtained in brain disorders.

Businesses must do a few things for sentiment analysis to be accurate. Sentiment or emotion analysis can be complex in NLP because machines should be trained to evaluate and understand sentiments as the human brain does. Finding and extracting sentimental data from social network posts to collect social users' behavior and opinions needs particular focus. Every phrase may include multiple scenarios and their polarities (i.e., positive and negative). The correct polarity of a sentence is necessary to mine various sentence circumstances. Present methods extract only one element at the text and sentence levels, which is needed to accurately convey the sentimental data within the sentence. A decision-based deep-Recurrent Neural Network (D-RNN) is introduced in this paper to find accurate sentiment analysis on several datasets, including Twitter, Restaurants, and laptops.

II. CONTRIBUTION

This work is mainly focused on extracting the efficient sentiment analysis for several datasets such as Twitter, Restaurants, and Laptops.

- ➢ The proposed work combines several text mining and sentiment analysis algorithms to effectively get the sentiments from the given datasets.
- ➢ An effective pre-trained BERT-Large cased model is used to train the sentiment datasets. The performance of training is to be improved by adopting the Stochastic Gradient Descent (SGD), an optimization algorithm. SGD effectively fine-tunes the pre-trained model to get accurate results.
- ➢ Better pre-processing techniques focused on data cleaning, tokenization, stop-words removal, and removing special characters from the given reviews.
- ➢ To extract better features, feature extraction techniques such as Bag of Words (BoW) and word embeddings such as Word2Vec are used to capture the semantic and contextual meaning of words and are widely used in NLP tasks, including sentiment analysis in multiple domains.
- ➢ The better classification of sentiment words is done by using the Deep sentiment analysis (DSA) based on Aspect Based Sentiment Analysis (ABSA) AND Priority Based Sentiment Analysis (PBSA) with Decision-based Recurrent Neural Network (D-RNN) yields better results that aid in improving business analysis. The decision-making model helps to make better decisions based on the output layer in RNN.

III. LITERATURE SURVEY

Gu et al. [12] proposed an innovative model called MBGCV, which classifies the sentiments from the text reviews. MBGCV is an integrated approach that combines several models, such as BiGTU, CNN, and VIB models. High-level sentiment features are extracted from the input dataset to find accurate sentiment patterns. Hayat et al. [13] mainly focused on solving sentiment analytic issues and enhancing performance in terms of sentiment score. Liu. [14] Explains the better sentimental-based lexicon models compared with ML and DL approaches. These models focused on addressing and sorting the issues related to accuracy detection. Thus the proposed model obtained a better sentiment score. Gupta et al. [15] introduced a lexicon-based model (LBM) classification approach based on corona virus tweets. The Twitter data was analyzed based on drugs, instances, and circumstances across the patients' ideas during lockdown time. LBM identifies the positive and negative reviews based on the situations handled by the Government of India.

TABLE 1. Types of sentiment analysis algorithms with performances.

Authors	Proposed Model	Dataset	Performance Metrics
A. A. Aziz et al., [34]	Contextual Analysis (CA)	Amazon	Acc-80.45%
A. Deniz et al., [35]	A binary classification model	Benchmark dataset, Stanford Sentiment Treebank	Acc-84.5%
L. Yang et al., [36]	Sentiment Lexicon and Deep Learning	dangdang.com	Acc-86.78
M. U. Salur et al., [37]	A novel hybrid deep learning Model	Turkish tweets	Acc-82.67
J. Du et al., [38]	Convolution-based attention model	Twitter Data	Acc-86.45
F. Yin et al., [39]	sentiment lexicon based on context-dependent part-of-speech (POS) chunks	Long text review dataset (LMRD)	Acc-82.45
S. Hu et al., [40]	credibility, interest and sentiment-enhanced recommendation (CISER) model	Amazon Camera review dataset	Precision-87.56
A. Rasool et al., [41]	Hybrid Sentiment Classification	Twitter Dataset	Accuracy-92
T. Bhattacharyya et al., [42]	Mayfly-Harmony Search (MA-HS)	high-dimensional microarray datasets	Acc-90.23
Y. Hu et al., [43]	Multi-objective particle swarm optimization	UCI datasets	Acc-89.89

For better outcomes, the linear SVC is combined with LBM. Elouardighi et al. [16] proposed the integrated approach by combining N-grams with TF-IDF. The proposed method analyzes the comments gathered from SNS such as Facebook, which belongs to the Arabic language. The information pertains to Morocco's Legislative Elections held in 2016. The comparison between NB, RF, and SVM is shown by the author. Vyas et al. [17] developed the sentiment-based model belonging to Coronavirus. The classification of tweets is totally based on opinions and reviews given by users from various OSN platforms. Khan et al. [18] proposed the deep LSTM technique which estimates opposite opinions and sentiments using the Kaggle sentiment dataset. The accuracy of D-LSTM is 91.32%, which is higher than existing models. Imran et al. [19] proposed the D-LSTM to estimate the sentiments extracted from the Kaggle Twitter dataset. The proposed model mainly analyzes the sentiments from the emoticons by extracting unique features from the Twitter data.

Several DL models for sentiment analysis are described by Antonakaki et al. [20]. The author classified the topics belongs to fake news, spam data and dangerous messages. From the result analysis the sentiment are identified based on the evaluation. The LSTMVIS model, proposed by Strobelt et al. [21], processes complicated patterns observed in many domains. Kumar et al. [22] introduced an integrated recommended system (IRS) for the movie dataset. The model suggested that IRS obtained better results combined with CF and CBF. The proposed method investigated current trends, public sentiment, and user responses. Bhatia [23] introduced a new graph model for identifying replicate terms that used graphs to correct the words. PCA is used to categorize text and shrink dimensions. The proposed method resulted in more accurate sentiment-based opinion-mining results. Davis et al. [24] proposed various ML models that can analyze various studies based on E-commerce datasets. The comparisons between several models give better outcomes over multiple review datasets. Tayal et al. [25] proposed a combined model based on various functionalities including pre-processing and feature extraction that helps to remove the noise from the input dataset. Semantic Sentence Similarity (SSS) combined with n-gram approach connected with particular sentences. The evaluation result shows better performance in terms of given metrics. Aslanian et al. [26] introduced an improved hybrid recommender system (HRS). The HRS addressed the cold-start issue, the feature connection matrix, and collective filtering. Better performance is achieved in terms of accuracy compared with existing ones. Cambria [27] demonstrated an automated method of analyzing emotional sentiments. The proposed model achieved better emotions and reviews. Du et al. [28] introduced classification models that process the sentiments from the two datasets. Advanced feature extraction with a softmax classifier gives a better classification over sentiment analysis. Giatsoglou et al. [29] describe the fast sentiments extracted model that finds emotions over different languages. The training is given with the polarity classification model that evaluates the performance of the proposed approach. The model performance is tested with four datasets gathered from Kaggle. Tan et al. [30] introduced the LDA model that extracts accurate sentiments based on the "popularity" reviews. An integrated model is used to explain the variations in sentiments. Jianqiang and Xiaolin [31] proposed an advanced preprocessing technique that removes the noise from the given datasets. It is also focused on feature extraction and classification of sentiment words. Tang et al. [32] proposed an integrated system that classifies the sentiments at the sentence level. To determine the correct polarity of a sentence, it is necessary to extract and analyze various contextual factors. This model classifies the sentiments based on the segmentation of the ranking model to estimate the polarity score. Sohrabi et al. [33] proposed an effective opinion-mining preprocessing method that will analyze feedback from users on the social network Twitter. Various preprocessing methods were applied to the dataset to achieve an acceptable standard text. The fast and accurate Word2vec method was also used for converting the word arrays to mathematical vectors. Following this quick and precise preprocessing phase supervised learning machine-learning techniques were applied to the acquired data.

IV. PROPOSED METHODOLOGY

This section explains the methodology that focuses on extracting accurate sentiments from the three datasets: Twitter, Restaurant, and Laptop. The proposed Deep sentiment analysis (DSA) method follows a step-by-step process. The first step focuses on the pre-trained model Bert-large-cased (BLC), consisting of 24-layer, 1024-hidden, 16-heads, 340M parameters. The second step follows data preprocessing, such as removing the noise, such as particular characters, URLs, etc. There are two feature extraction techniques such as Bag of Words (BoW) and word embeddings, such as Word2Vec used to extract the accurate features that show a significant impact on outputs. The proposed approach DSA combines ABSA and PBSA with a Decision-based Recurrent Neural Network (D-RNN). Figure 1 explains the step-by-step process for sentiment analysis using DSA.

A. REMOVING SPECIAL CHARACTERS FROM THE GIVEN REVIEWS

In this step, the substitution function detects the special characters alphanumeric (i.e., not letters or numbers) with spaces. The regular expression [^\w]+ matches any character that is not in the range of 0-9, a-z, A-Z, and underscores. It also replaces the string with space in this case. The resulting string is then stored back in the review variable.

For Example: Input: "@HeathCastor dratz oh well, i uphold you to atleast have SOME scruff to be watson"

Output: HeathCastor dratz oh well, i uphold you to atleast have SOME scruff to be Watson.

B. TOKENIZATION

In this step, the tokenization is dividing a text into units known as tokens, which can be words, punctuation marks, numbers, or other meaningful elements. Tokenization is a necessary step in sentiment analysis, which analyses the text to determine whether it is positive, negative, or neutral. Each score is between 0 and 1. The negative score represents the −1 to −5, 0-5 neutral and 5-10 positive.

The following equation used for measuring the tokenization for sentiment analysis:

$$\text{Score} = \sum (\text{weight}_i * \text{value}_i) \tag{1}$$

where, every word in the text or review is initialized a weight based on its importance which specifies the sentiment. The following steps involves in preprocessing:

Remove any punctuation and convert to lowercase: "HeathCastor dratz oh well, i uphold you to atleast have SOME scruff to be Watson"

Split the text into individual words: ["health", "castor", "dratz", "oh", "well", "i", "uphold", "you", "to", "atleast", "have", "some", "scruff", "to" "be" "watson"]

Remove stop words: ["heatlh", "castor", "dratz", "uphold", "atleast", "scruff", "watson"]

Assign weights to each word based on its sentiment: {"heatlh", "castor", "dratz", "uphold", "atleast", "scruff", "watson"}

"health": 0.5, "castor": 0, "dratz": −0.25, "uphold": 0.25, "atleast": 0, "scruff": −0.25, "watson": 0

Here, the weights assigned are 0.5 to the word "health", which has the highest positive sentiment score of +3, indicating a very positive sentiment. Similarly, for the other words the weights are assigned as −0.25 to the words "dratz" and "scruff", which have the lowest negative sentiment score of −2, indicating a very negative sentiment. Words with a sentiment score of 0, such as "castor", "atleast", "some", and "watson", are assigned a weight of 0, indicating a neutral sentiment. Finally, the word "uphold" is assigned a weight of 0.25, which is a moderate positive sentiment score of +2.

Sum the values of all the words to get the sentiment score: 0.5 + 0 - 0.25 + 0.25 + 0 –0.25 + 0 = 0.75.

The overall sentiment score for this text is 0.75, which is a neutral sentiment. These steps are applicable for all the reviews belong to datasets.

C. BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT-LARGE CASED)

BERT-large cased is the pre-trained model that utilizes a transformer-based system to create inspected word embeddings. The performance of proposed model improved with fine-tuned embeddings to process the complex sentiments.

Here are the equations for BERT using sentiment analysis:

1) WORD EMBEDDING LAYER

The input sentence is tokenized and every token is summarized to a corresponding vector representation utilizing an embedding matrix. The embedding matrix can be represented as follows:

$$\mathrm{E} = [\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{n}] \tag{2}$$

n-represents the no of tokens in the sentence, for i^{th} token three$_i$ represents the embedding vector.

2) TRANSFORMER ENCODER LAYER

The embeddings are then fed into a multi-layer two-way transmission encoder. This layer contains N uniform layers; every layer contains multi-head self-attention mechanism and a FF-NN. The output of the i^{th} layer can be represented as:

$$\mathrm{H}_i = \text{TransformerEncoderLayer}\,(\mathrm{H}_i - 1) \tag{3}$$

where H_0 is the input embeddings and H_N is the last hidden layer represents the sentence.

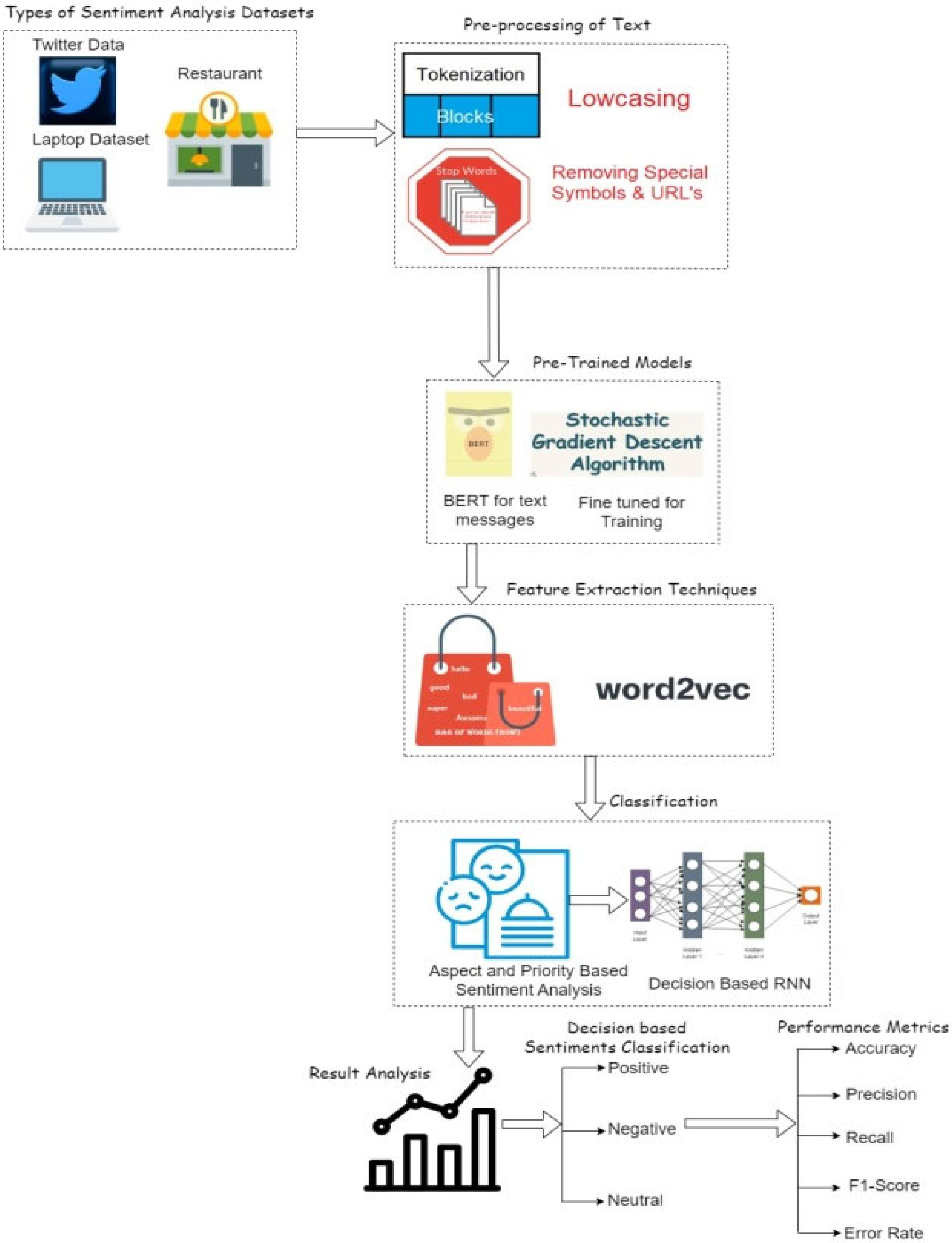


FIGURE 1. System architecture.

3) POOLING LAYER

A pooling layer obtains a fixed-sized representation of the entire sentence. The output of the last transformer encoder layer, H_N, is fed into a pooling layer that aggregates the words across all tokens in the sentence. One commonly used pooling strategy is mean pooling, which represents as follows:

$$P = \text{Mean}\,(H_N) \tag{4}$$

where P is the pooled represents the sentence.

4) FULLY CONNECTED LAYER (FCL)

Finally, the pooled representation is passed through a FCL with a softmax activation function to predict the sentiment of the sentence. The result of the FCL can be given as:

$$y = \text{softmax}\,(W_0 * P + b_0) \tag{5}$$

where W_0 and b_0 are the weight and bias parameters of the fully connected layer, and y is a vector of probabilities representing the sentiment of the sentence.

D. STOCHASTIC GRADIENT DESCENT (SGD) IN SENTIMENT ANALYSIS

SGD is a well-known optimization algorithm used in machine learning to train models. SGD's basic idea is to iteratively adjust a model's parameters to minimize the loss function, which measures how well the model performs on a given task [44]. The goal of sentiment analysis is to train a model that can categorize text as positive, negative, or neutral. The model accepts a string of words as input and returns a probability distribution across three classes. In sentiment analysis, the information for SGD is a set of labeled training data that contains text messages and their associated sentiment labels (e.g., positive, negative, or neutral).

The cross-entropy is a loss function for sentiment analysis. Consider a small subset of the training data (a "mini-batch"). Assume we have labeled reviews; each assigned a sentiment label (positive, negative, or neutral). Let X denotes set of opinions, and 'y'represents the accompanying labels. Let f(x;) denote our model's output for input x, where denotes the model parameters. It can be measured by using: (6)

$$L(\theta) = -\frac{1}{m} * \sum (y * \log(f(x;\theta)) + (1-y) * \log(1 - f(x;\theta))) \tag{6}$$

where m is the total number of reviews, Σ denotes the sum over all reviews, and log denotes the natural logarithm. The first term in the sum represents the loss for positive reviews, while the second term represents the loss for negative reviews.

SGD can optimize model parameters θ by iteratively updating them concerning the gradient of the loss function. It can be calculated using the following formula:

$$\nabla L(\theta) = \frac{1}{m} * \sum ((f(x;\theta) - y) * x) \tag{7}$$

where $*$ denotes element-wise multiplication. The update rule for SGD can be written as:

$$\theta = \theta - \alpha * \nabla L \tag{8}$$

where α is the learning rate, which controls the step size of the update. By repeatedly applying the above update rule, we can minimize the loss function and learn a model that can predict the sentiment of a given review. The output of SGD in sentiment analysis is a trained model that can predict the sentiment of new text samples. The model's accuracy in predicting the correct sentiment label is typically evaluated on a separate set of labeled data known as the test set.

E. FEATURE EXTRACTION

Feature extraction plays the significant role in sentiment analysis. In this paper, the two feature extraction techniques are used such as BOW and Word2Vec

1) BAG-OF-WORDS (BOW) MODEL

The BOW model represents a text as a collection of words and their frequency counts. For sentiment analysis, the frequency of positive and negative words can be used to determine the sentiment of a text. Here's the equation:

$$\text{Positive Sentiment Score } (P_s) = \frac{(\text{Number of Positive Words})}{(\text{Total Number of Words})} \tag{9}$$

$$\text{Negative Sentiment Score } (N_s) = \frac{(\text{Number of Negative Words})}{(\text{Total Number of Words})} \tag{10}$$

2) Word2Vec

Word2Vec is a popular feature extraction technique in natural language processing tasks like sentiment analysis. It is a neural network-based method for representing words as vectors in a high-dimensional space, with each dimension representing a distinct word feature. The Word2Vec model is trained on a large corpus of text data in sentiment analysis to learn the relationship between words and their context. It allows capturing the underlying semantic meaning of words, which helps predict text sentiment.

Here's an example of how Word2Vec can be used for feature extraction in sentiment analysis:

Suppose we have a dataset of laptop reviews of a product, and we want to predict whether each review is positive or negative. It starts with removing stop words, unique case letters, and lowercase letters and tokenizing the text into individual words. Next, we use the Word2Vec model to generate a feature vector for each word in the dataset. For example, the word "good" might be represented as a vector with values [0.2, 0.1, 0.5, ...], where each value corresponds to a specific feature of the word. We can then use these feature vectors to represent each review as a numerical vector, by taking the average of the feature vectors of all the words in the review. For example, if a review contains the words "good" and "product", we can represent it as a vector with values [0.15, 0.05, 0.4, ...], which is the average of the feature vectors of "good" and "product". Finally, we can train a machine learning model such as logistic regression or a neural network on the vectorized reviews, using the corresponding sentiment labels as the target variable. The trained model can then be used to predict the sentiment of new, unseen reviews.

Overall, Word2Vec provides an effective way to capture the underlying meaning of words and to represent text data as numerical vectors, which can be used as input features for ML models in sentiment analysis.

F. ASPECT-BASED SENTIMENT ANALYSIS (ABSA)

ABSA is a subset of NLP that focuses on identifying the sentiments expressed in a text about different elements or attributes of a product or service [45] [46]. ABSA involves identifying the aspects or entities mentioned in the text, extracting the sentiment associated with each aspect, and then adding the sentiment scores to get the overall sentiment score for the text [47]. The sentence contains an aspect of the corresponding output value if that aspect exceeds the threshold. In this paper, the advanced features were extracted using the

Bag-of-words (BoG) and Word2Vec, which help improve the aspect-based sentiment score. ABSA combined with RNN shows practical sentiment analysis by reducing the error loss—the input of the ABSA taken from the feature extraction, which consists of BoG and Word2Vec. The formula for measuring the sentiment score is given in Equation (11), as shown at the bottom of the page.

Example: Consider a review of a restaurant: "The food was great, but the service was terrible."

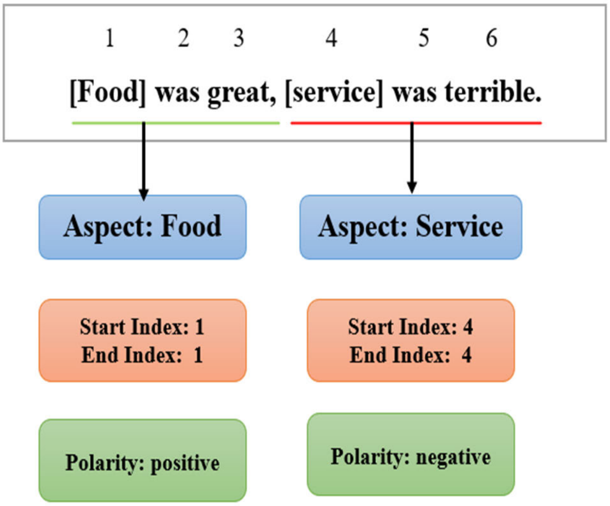


FIGURE 2. An illustration of the use of aspect-based sentiment analysis.

G. ASPECT EXTRACTION

In this step, ABSA is to identify the aspects or features that are being evaluated. In this example, the aspects are "food" and "service".

H. SENTIMENT EXTRACTION

Next, the sentiment associated with each aspect needs to be extracted. A sentiment lexicon contains a list of words and their associated sentiment scores can be used to accomplish this. In this example, let's assume that the sentiment lexicon assigns the following scores:

"great": +1

"terrible": −1

Using these scores, we can calculate the sentiment for each aspect as follows:

Sentiment(food) = +1

Sentiment(service) = −1

I. OVERALL SENTIMENT

Finally, the combined sentiment scores for every aspect are used to obtain the review's overall sentiment score. One way to do this is to use a weighted average, where the weights are the importance of each aspect. In this example, let's assume that the importance weights are equal:

$$\begin{aligned} \text{TotalSentiments (TS)} &= ((\text{Sentiment(food)} \\ &\quad + \text{Sentiment(service)})/2 \\ &= (+1-1)/2 \\ &= 0 \end{aligned}$$

- ➢ The overall sentiment score is zero, which indicates a neutral sentiment given in above equation.
- ➢ In summary, aspect-based sentiment analysis involves identifying the aspects or features being evaluated; extracting the sentiment associated with each aspect using a sentiment lexicon, and merges the scores of the sentiments to get total score for sentiments.

Opinion Aggregation: Aggregate the sentiment scores for each aspect to determine an overall sentiment score for that aspect.

Visualization and Reporting: Visualize the results and report the findings, including the most positive and negative aspects of the product/service.

ABSA is a complex and difficult task in general, but the algorithm above provides a high-level overview of the process involved in analyzing opinions and sentiments about particular components or attributes of an item or service. The process of identifying and extracting emotional data from text is known as SA.It can be used to determine whether a piece of text has a positive, negative, or neutral sentiment. Here are some common equations used for sentiment analysis:

J. PRIORITY BASED SENTIMENT ANALYSIS (PBSA)

PBSA is a type of sentiment analysis that assigns a priority or weight to certain keywords or phrases within a text to determine the overall sentiment of the text [48]. In this method, specific words or phrases that are considered to be more important or impactful are given a higher priority, and their sentiment is weighed more heavily in determining the overall sentiment of the text.

$$\text{Priority Weighting} = \frac{(\text{Priority}_{\text{Score}}\ \text{-Minimum Priority}_{\text{Score}})}{(\text{Maximum Priority}_{\text{Score}} - \text{Minimum Priority}_{\text{Score}})} \tag{12}$$

For example, consider the following sentence:

"I really enjoyed the movie, but the ending was disappointing."

In this sentence, there are two contrasting sentiments. The first part of the sentence expresses a positive sentiment ("I really enjoyed the movie"), while the second part expresses a negative sentiment ("the ending was disappointing"). In priority-based sentiment analysis, the word

$$\text{Sentiment Score } (S_s) = \frac{(\text{Positive Sentiment Count } (P_c) - \text{ Negative Sentiment Count } (N_c))}{\text{Total Sentiment Count } (T_c)} \tag{11}$$

"disappointing" would be given a higher priority or weight, as it is a more impactful word that carries more weight in determining the overall sentiment of the sentence. Therefore, the overall sentiment of the sentence would be considered negative.

Priority-based sentiment analysis can be useful in cases where certain words or phrases carry more weight or importance in determining the overall sentiment of a text. This method allows for a more nuanced and accurate analysis of sentiment, as it takes into account the relative importance of different words and phrases within a text. The priority is measured by using the equation (12) calculates the priority weighting of a sentiment based on its priority score, which reflects the importance or relevance of the sentiment to the analysis. The priority weighting is normalized to a value between 0 and 1, based on the minimum and maximum priority scores in the dataset.

V. CLASSIFICATION OF SENTIMENT SCORES

Decision-based Recurrent Neural Networks (D-RNNs) is generally used for sentiment analysis tasks, as they are well-suited for processing sequential data. Decision making is the sub-field in machine learning domain which gives the efficient decisions based on outcomes. The algorithm steps of an RNN for sentiment analysis can be described as follows:

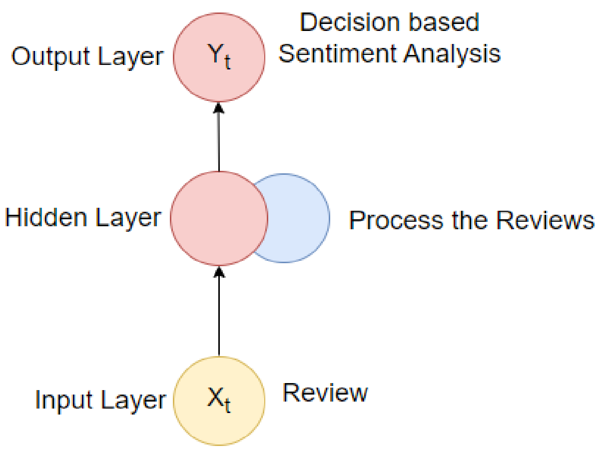


FIGURE 3. Structure of RNN.

Step 1: The input layer of the RNN receives the input sequence, which contains a continuous word embeddings representing the words in a sentence using equation (2).

Step 2: The hidden layer of the RNN is responsible for capturing the context of the input sequence. This is accomplished by keeping a hidden state vector that is modified at every phase according to the input data and the past hidden state. The equations governing the hidden layer are:

$$\begin{aligned} Z_t &= W_z X_t + U_z h_{t-1} + b_z r_t = W_r X_t + U_r h_{t-1} + b_r r_t \\ &= (1 - z_t) * h_{t-1} + z_t * \tanh(W_h X_t \\ &\quad + U_h r_{t-1} * r_t + b_h \end{aligned} \tag{13}$$

Here, Z_t and r_t are the update and reset gates, respectively. They decide the quantity of the previous hidden state will remain same, and the current input will be utilized for modifying the current hidden state. The present hidden state vector is h_t, the input vector at time t is x_t, and the weighted matrices W and U are acquired throughout training. b denotes a bias vector.

Step 3: The output layer of the RNN takes the final hidden state vector as input and produces a sentiment score.

COMBINING THE ASPECT BASED SENTIMENT ANALYSIS (ABSA) AND PRIORITY BASED SENTIMENT ANALYSIS (PBSA) WITH RECURRENT NEURAL NETWORK (RNN)

ABSA and PBSA are two different approaches used to analyze the sentiment of a text. ABSA focuses on identifying the sentiments over particular aspects or entities in the given inputs, while PBSA prioritizes the sentiments of the overall text present on predefined priorities.–

To combine ABSA and PBSA using a recurrent neural network (RNN), the following algorithm steps are used:

Step 4: Let x be the input text, and a be the aspect or entity we want to analyze the sentiment towards.

$$h_{absa} = \mathrm{RNN}\,(x) \tag{14}$$

//Represented the ABSA equation

$$y_{absa} = \mathrm{Softmax}\,(W_{absa} * h_{absa}) \tag{15}$$

where h_{absa} is the hidden state of the RNN, y_{absa} is the output sentiment score for the aspect a, and W_{absa} is the weight matrix.

Step 5: Let x be the input text, and p be the set of predefined priorities.

$$h_{pbsa} = \mathrm{RNN}\,(x) \tag{16}$$

// Represented the PBSA equation

$$y_{pbsa} = \mathrm{Softmax}\left(W_{pbsa} * h_{pbsa}\right) \tag{17}$$

where h_{pbsa} the hidden state of the RNN, y_{pbsa} is the output sentiment score for the overall text, and W_{pbsa} is the weight matrix.

Step 6: Combining ABSA and PBSA:

To combine ABSA and PBSA, we can use a weighted sum of the ABSA and PBSA output scores:

$$y_{combined} = \text{alpha} * \text{y\_absa} + (1 - \text{alpha}) * \text{y\_pbsa} \tag{18}$$

where $y_{combined}$ is the combined sentiment score, and alpha is a weighting factor that determines the relative importance of ABSA and PBSA.

The final sentiment label can be determined based on the value of $y_{combined}$

VI. IMPLEMENTATION DETAILS

To implement all these models a high configuration system is used. The system configuration is Windows 11th generation with windows 10 operating system, 32 GB DDR4 RAM and 32 GB GPU. Python IDLE used for performance analysis and used the libraries such as Keras, pandas, sklearn and matplotlib. The hyper-parameters shown in table 2.

TABLE 2. Hyper-parameters used in this bert-large cased and optimized SDG with D-RNN.

Name of the Hyper-parameter	Value
Batch size	20
Learning rate	0.001
Hidden layers	N
Number of Epochs	50
Optimizers	SGD
Activation	Softmax

TABLE 3. Shows the training and testing of three datasets.

Datasets		Positive (pos)	Negative (Neg)	Neutral (Neu)	Tot al	Attrib utes	Source
Laptop-ACOS	Training	986	567	447	2000	11	Kaggle
	Testing	867	534	675	2076	11	
Restaur ant	Training	789	345	366	1500	2	Kaggle
	Testing	698	369	433	1500	2	
Twitter	Training	2698	1456	846	5000	6	Kaggle
	Testing	2589	1596	815	5000	6	

A. DATASETS DESCRIPTION

Three datasets, Twitter [49], Restaurants, and Laptop-ACOS [50], were used and collected from Kaggle. The experiments are conducted by using Python programming language. Every dataset consists of two folders such as training and testing. All these datasets reflect the reviews the customers give belonging to sentiment analysis. Table 3 shows the detailed view of datasets. The preprocessing techniques are applied to these datasets, removing the stop-words, special characters, URLs, and punctuations. The twitter dataset contains 162980 tweets. By using the API crawler a unique 10k tweets are extracted from total available tweets that are uses for the experiments.

B. RESULTS AND DISCUSSIONS

The training is focused on BERT-large cased combined with optimized model SGD to improve the performance model. SGD is the fine tuning model that improves the performance of BERT-large cased by training all the features of the given datasets. Python programming language is used for the development of training models. A better hardware is required for the system configuration like 16 GB RAM with 2 GB NVIDIA graphic card. The training set consists of 50% and testing set 50% of text data approximately I,e as shown in the table 3 the laptop-ACOS contains 2000 reviews for training and 2076 reviews for testing. The same repeated for other datasets. The training loss represents the model's compatibility that fits into the training data. For each dataset, 50 Epochs are selected for five epochs; the training is completed. This section includes training and testing accuracy, training loss, and testing loss for TD-LSTM, LEAN, and Deep Sentiment Analysis (DSA). The training accuracy for TD-LASTM is 71.23%, and the testing accuracy is 72.34%; for LEAN, it is 81.56%, and the testing accuracy is 82.45%.

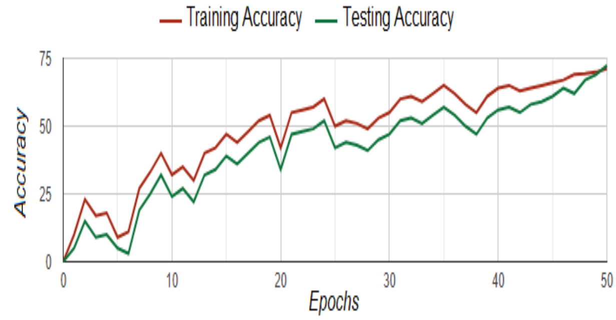


FIGURE 4. Training and testing accuracy for TD-LSTM.

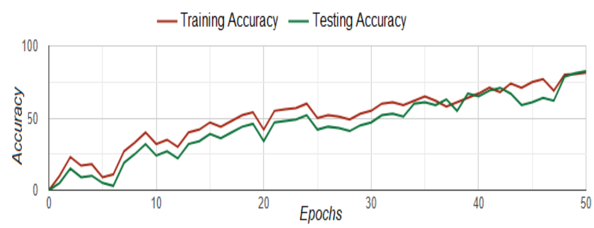


FIGURE 5. Training and testing accuracy for LEAN.

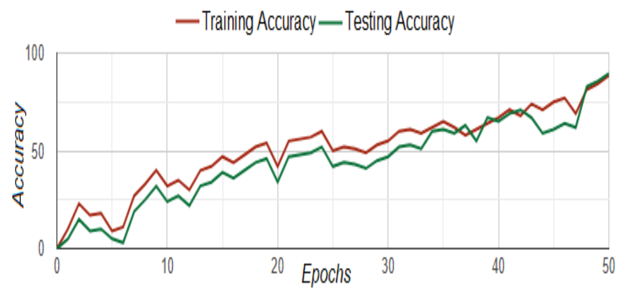


FIGURE 6. Training and testing accuracy for DSA.

FIGURE 7. Training and testing loss for TD-LSTM.

Finally, our proposed model obtained an accuracy of 88.98% and a testing accuracy of 89.56%. The training and testing loss for TD-LSTM is 0.008 and 0.14; for LEAN, it is 0.006, and the testing loss is 0.11; and finally, our proposed model achieved a loss is 0.003, and a testing loss is 0.10. Thus the training and testing model performs better for all three datasets.

FIGURE 8. Training and testing loss for LEAN.

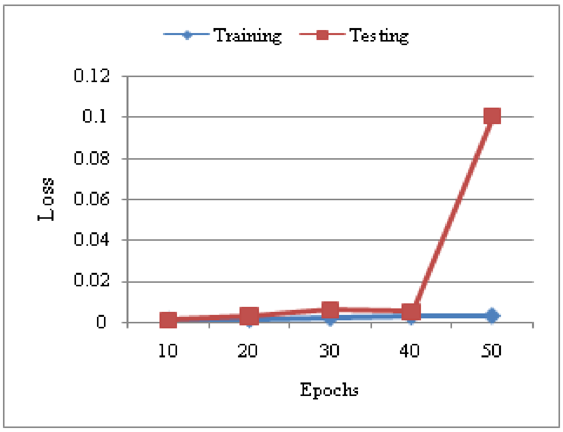


FIGURE 9. Training and testing loss for DSA.

VII. PERFORMANCE METRICS

The confusion matrix measures the performance of the proposed model based on the following instances explained in figure 9. The given metrics can help us evaluate the model's strengths and weaknesses and decide whether to adjust the model or collect more data.

$$\text{Accuracy } (A_{cc}) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision } (P_{rec}) = \frac{TP}{TP + FP}$$

$$\text{Recall } (R_{call}) = \frac{TP}{TP + FN}$$

$$F1 - \text{Score } (F1S) = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Error Rate}(E_{error}) = 1 - \text{Accuracy}$$

Table 4 shows the instances of laptop-ACOS dataset by using the confusion matrix. A comparative performance of existing models compared with proposed DSA. The proposed DSA achieved the high performance in terms of TP, TN, FP and FN. Figure 9 shows the visualization of obtained results by using the confusion matrix instances. Figure 10 shows the overall count values obtained from the confusion matrix for the classes positive, negative and neutral. The positive and neutral are considered as one class and negative considered as negative.

TABLE 4. Confusion matrix for Laptop-ACOS dataset.

Algorithms	TP	TN	FP	FN
Bi-RNN	678	489	453	453
Bu-GRU	734	567	418	357
TD-LSTM	872	676	176	276
LEAN	934	789	164	189
DSA (Proposed)	1101	890	53	32

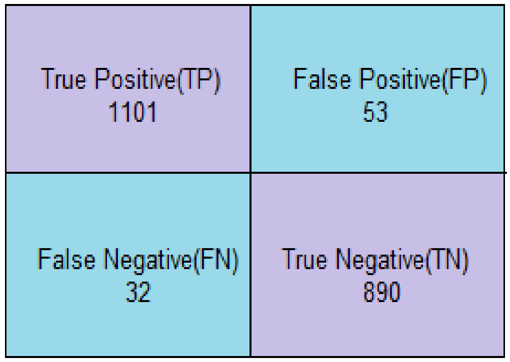


FIGURE 10. Confusion matrix count for Laptop-ACOS using DSA (proposed).

TABLE 5. Comparative performances of several algorithms for Laptop-ACOS dataset.

Algorithms	A_{cc}	P_{rec}	R_{call}	F1-Score	E_{error}
Bi-RNN	0.563	0.5995	0.5995	0.5995	0.437
Bu-GRU	0.6267	0.6372	0.6728	0.6575	0.3733
TD-LSTM	0.774	0.832	0.7596	0.7942	0.226
LEAN	0.83	0.85	0.8317	0.841	0.17
DSA (Proposed)	0.959	0.954	0.9718	0.9628	0.041

Table 5 shows the overall comparative performances of the existing models and proposed DSA model based on performance metrics. The DSA achieved the accuracy of 0.959, precision 0.954, recall 0.9718, F1-score 0.9628 and error rate 0.041. There is a solid difference between existing and proposed models and the average difference between every model is approximately 7.6%. The error rate is low for DSA compare with other existing models. Figure 11 shows the visualization of comparative performances between the existing and proposed models.

Table 6 and figure 12 shows the detailed performance in terms of sentiment analysis such as positives, negatives and neutrals. These sentiments are analyzed from the Laptop-ACOS dataset. The overall positives, negatives and neutrals are shown in table 6 and figure 12. The proposed model DSA

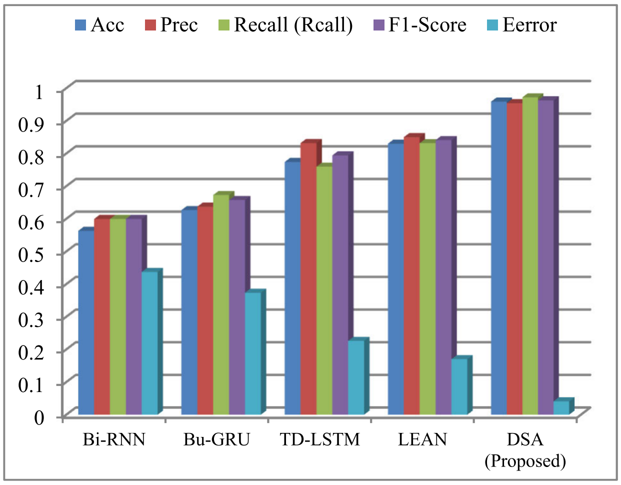


FIGURE 11. Comparative performances of several algorithms for Laptop-ACOS dataset.

TABLE 6. Overall comparative performances of several algorithms for Laptop-ACOS dataset.

Algorithms		A_{cc}	P_{rec}	R_{call}	F-S-Score	E_{error}
Bi-RNN	Positive	0.534	0.567	0.574	0.545	0.466
	Negative	0.545	0.557	0.564	0.551	0.455
	Neutral	0.538	0.562	0.565	0.542	0.462
Bu-GRU	Positive	0.6231	0.602	0.6723	0.6576	0.3769
	Negative	0.632	0.604	0.6645	0.6587	0.368
	Neutral	0.636	0.617	0.6573	0.6523	0.364
TD-LSTM	Positive	0.79	0.8	0.798	0.784	0.21
	Negative	0.8	0.798	0.812	0.785	0.2
	Neutral	0.79	0.782	0.79	0.79	0.21
LEAN	Positive	0.83	0.843	0.846	0.851	0.17
	Negative	0.87	0.871	0.878	0.879	0.13
	Neutral	0.887	0.876	0.884	0.883	0.113
DSA (Ours)	Positive	0.964	0.961	0.961	0.963	0.036
	Negative	0.956	0.953	0.956	0.962	0.046
	Neutral	0.963	0.961	0.968	0.969	0.037

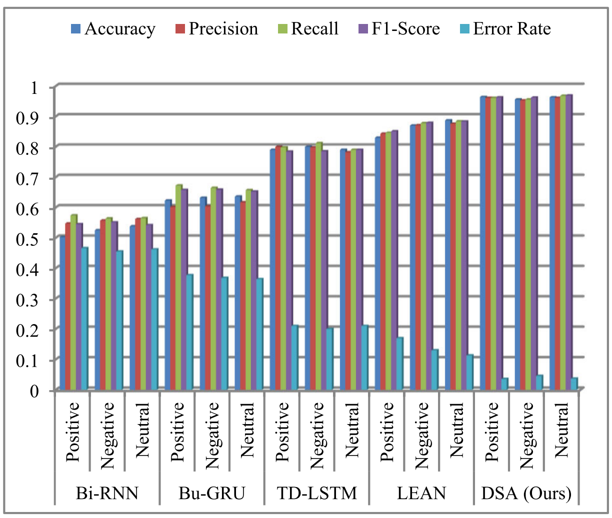


FIGURE 12. Overall comparative performances of several algorithms for Laptop-ACOS dataset.

detected the total positives, negatives and neutral based on the sentiment analysis. The error rate is also low for DSA, representing fewer errors from the dataset.

TABLE 7. Confusion matrix for restaurant dataset.

Algorithms	TP	TN	FP	FN
Bi-RNN	462	354	389	295
Bu-GRU	532	398	351	216
TD-LSTM	623	456	256	165
LEAN	731	541	178	80
DSA (Proposed)	1134	231	147	88

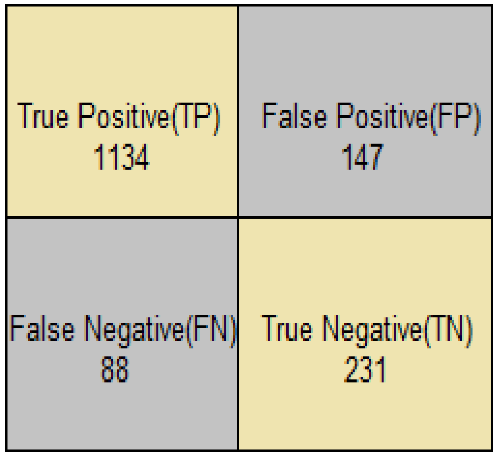


FIGURE 13. Confusion matrix count for restaurant dataset using DSA (proposed).

TABLE 8. The comparative performances of several algorithms for restaurant dataset.

Algorithms	A_{cc}	P_{rec}	R_{call}	F-S-Score	E_{error}
Bi-RNN	0.5047	0.5429	0.5662	0.5543	0.4953
Bu-GRU	0.6212	0.6025	0.7112	0.6524	0.3788
TD-LSTM	0.6856	0.7088	0.7906	0.7475	0.3144
LEAN	0.828	0.8042	0.9014	0.85	0.172
DSA (Proposed)	0.8531	0.8852	0.928	0.9061	0.1469

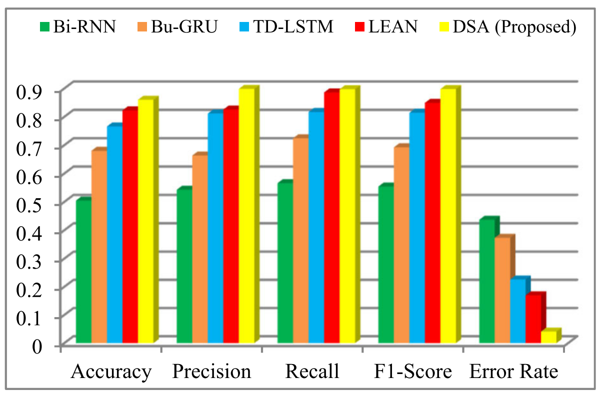


FIGURE 14. Comparative performances of several algorithms for restaurant dataset.

Table 7 shows the total TPs, TNs, FPs, and FNs obtained from the confusion matrix to analyze the model performance.

TABLE 9. The comparative performances of several algorithms for restaurant dataset.

Algorithms		A_{cc}	P_{rec}	R_{call}	F-S-Score	E_{error}
Bi-RNN	Positive	0.5023	0.547	0.574	0.545	0.4977
	Negative	0.525	0.557	0.564	0.551	0.475
	Neutral	0.538	0.562	0.565	0.542	0.462
Bu-GRU	Positive	0.6231	0.602	0.6723	0.6576	0.3769
	Negative	0.632	0.604	0.6645	0.6587	0.368
	Neutral	0.636	0.617	0.6573	0.6523	0.364
TD-LSTM	Positive	0.79	0.8	0.798	0.784	0.3144
	Negative	0.8	0.798	0.812	0.785	0.3029
	Neutral	0.79	0.782	0.79	0.79	0.3166
LEAN	Positive	0.83	0.843	0.846	0.851	0.17
	Negative	0.87	0.871	0.878	0.879	0.13
	Neutral	0.887	0.876	0.884	0.883	0.113
DSA (Ours)	Positive	0.964	0.961	0.961	0.963	0.133
	Negative	0.956	0.953	0.956	0.962	0.099
	Neutral	0.963	0.961	0.968	0.969	0.059

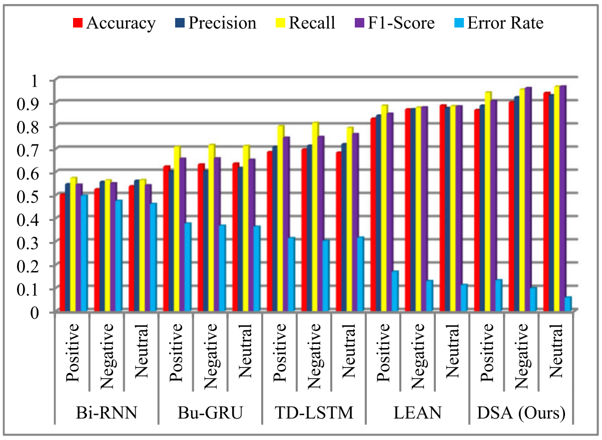


FIGURE 15. Overall comparative performances of existing and proposed sentiment analysis algorithms for restaurant dataset.

TABLE 10. Confusion matrix for twitter dataset.

Algorithms	TP	TN	FP	FN
Bi-RNN	1467	1557	897	1079
Bu-GRU	1754	1557	890	667
TD-LSTM	2565	1265	596	574
LEAN	2256	1456	510	290
DSA (Proposed)	3079	1221	348	352

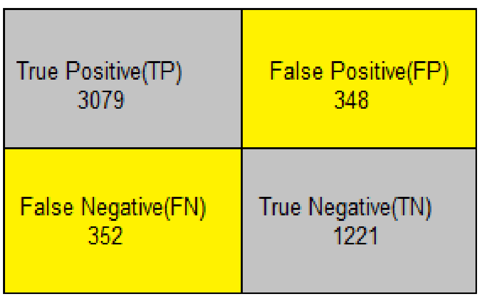


FIGURE 16. Confusion matrix count for twitter dataset using DSA (proposed).

For the Bi-RNN, the count values were very low; for DSA, the count values were high. The total TPs-462, TNs-354, FPs-389, and FNs-295 were obtained for Bi-RNN. The proposed model achieved high TPs-1134, TNs-231, FPs-147, and FNs-88. Conversely, the other existing models also poorly identified the TPs, TNs, FPs, and FNs. Figure 13 represents the confusion matrix obtained from table 7. Table 8 shows the overall performances of existing and proposed models based on the obtained results. The Bi-RNN achieved a low accuracy of 0.5047%, a high error rate of 0.4953%, and DSA achieved a high accuracy of 0.8531%. The average difference for every model is about 0.7%. The overall difference between all the models is shown in Figure 14.

TABLE 11. List of performances of several algorithms applied on twitter dataset.

Algorithms	A_{cc}	P_{rec}	R_{call}	F-S-Score	E_{error}
Bi-RNN	0.6048	0.6206	0.5762	0.5976	0.3952
Bu-GRU	0.6802	0.6634	0.7245	0.6926	0.3198
TD-LSTM	0.766	0.8115	0.8171	0.8143	0.234
LEAN	0.8227	0.8256	0.8861	0.8494	0.1773
DSA (Proposed)	0.86	0.8985	0.8974	0.898	0.14

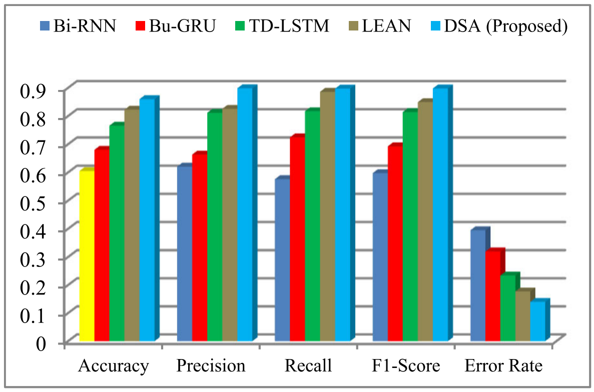


FIGURE 17. The comparative performances of several algorithms for Twitter dataset.

TABLE 12. The comparative performances of several algorithms for twitter dataset.

Algorithms		A_{cc}	P_{rec}	R_{call}	F-S-Score	E_{error}
Bi-RNN	Positive	0.604	0.623	0.574	0.597	0.396
	Negative	0.6178	0.618	0.564	0.598	0.3822
	Neutral	0.605	0.628	0.565	0.612	0.395
Bu-GRU	Positive	0.681	0.667	0.723	0.698	0.322
	Negative	0.678	0.634	0.734	0.678	0.368
	Neutral	0.685	0.656	0.731	0.697	0.315
TD-LSTM	Positive	0.76	0.81	0.821	0.823	0.24
	Negative	0.77	0.798	0.812	0.813	0.23
	Neutral	0.78	0.782	0.801	0.826	0.22
LEAN	Positive	0.83	0.843	0.846	0.851	0.17
	Negative	0.85	0.871	0.878	0.879	0.15
	Neutral	0.847	0.876	0.884	0.883	0.153
DSA (Ours)	Positive	0.912	0.912	0.903	0.939	0.088
	Negative	0.936	0.913	0.906	0.921	0.064
	Neutral	0.923	0.901	0.923	0.937	0.077

TABLE 13. Evaluation of the proposed model's efficiency over multiple datasets.

Model	Laptop					Restaurant					Twitter				
	Accuracy	Precision	Recall	F1-Score	Error Rate	Accuracy	Precision	Recall	F1-Score	Error Rate	Accuracy	Precision	Recall	F1-Score	Error Rate
Bi-RNN	0.563	0.5995	0.5995	0.5995	0.437	0.5047	0.5429	0.5662	0.5543	0.4953	0.6048	0.6206	0.5762	0.5976	0.3952
Bu-GRU	0.6267	0.6372	0.6728	0.6575	0.3733	0.6212	0.6025	0.7112	0.6524	0.3788	0.6802	0.6634	0.7245	0.6926	0.3198
TD-LSTM	0.774	0.832	0.7596	0.7942	0.226	0.6856	0.7088	0.7906	0.7475	0.3144	0.766	0.8115	0.8171	0.8143	0.234
LEAN	0.83	0.85	0.8317	0.841	0.17	0.828	0.8042	0.9014	0.85	0.172	0.8227	0.8256	0.8861	0.8494	0.1773
DSA (Proposed)	0.959	0.954	0.9718	0.9628	0.041	0.853	0.8852	0.928	0.9061	0.1469	0.86	0.8985	0.8974	0.898	0.14

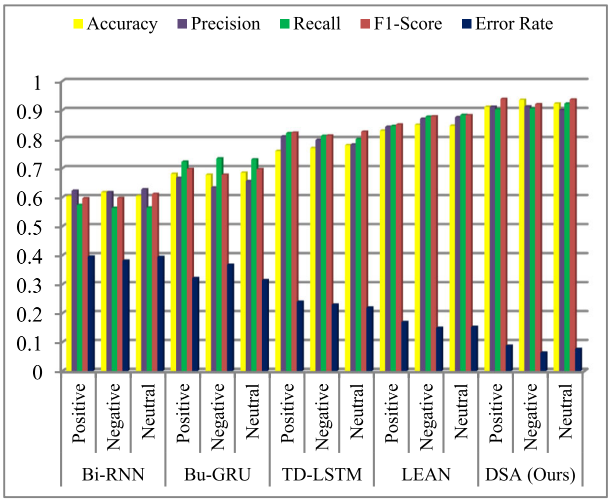


FIGURE 18. The comparative performances of several algorithms for Twitter dataset.

Table 9 shows the detailed performances of existing models Bi-RNN, Bu-GRU, TD-LSTM, and LEAN compared with DSA. The confusion matrix measures all the sentiments like positives, negatives, and neutrals. The proposed model DSA detected a more favorable rate of 0.954, a negative rate of 0.956, and the neutral rate of 0.963 compared with other existing models. The error rate is low for DSA, which is 0.133 for positives, negatives 0.099, and neutrals 0.059. The Bi-RNN achieved a low positive rate of 0.5023, a low negative rate of 0.525, and a low, neutral rate of 0.538. The difference between Bi-RNN and DSA is approximately 46.5%. Figure 16 shows a detailed view of the difference between all the models, such as Bi-RNN, Bu-GRU, TD-LSTM, and LEAN, compared with DSA.

Table 10 shows the performance of all the models based on count of TPs, TNs, FPs and FNs. These results are obtained for twitter dataset. The total testing set consists of 5000 reviews collected from Kaggle. The overall TPs count is 1467, TN 1557, FP 897 and FN 1079 for Bi-RNN which is very low compare with Bu-GRU, TD-LSTM and LEAN. The DSA count values are very high like TPs 3079, TN 1221, FP 348 and FN 352. Figure 16 shows the confusion matrix representation which is obtained from the confusion matrix count values.

Table 11 shows the overall performances of Bi-RNN, Bu-GRU, TD-LSTM, LEAN, and DSA (Proposed). The average difference between these models is 0.55%. Compared with LEAN, the proposed DSA model achieved an accuracy of 0.86, precision of 0.8985, recall of 0.8974, F1-score 0.898, and an Error rate is 0.14, which is low. The difference between DSA and LEAN is about 0.4%. Figure 17 shows the difference between all the models.

Table 12 shows the detailed performances of all models, such as Bi-RNN, Bu-GRU, TD-LSTM, LEAN, and DSA (Ours), based on the sentiment analysis, such as positive, negative, and neutral. The sentiment analysis performance is low for Bi-RNN and high for DSA. The Bi-RNN performance is positive at 0.604, negative at 0.6178, neutral at 0.605, and the Error rate at 0.396. The DSA performance in terms of positive is 0.912, negative is 0.936, neutral is 0.923, and the Error rate is 0.077. The DSA achieved high performance in terms of given performance metrics. Figure 18 shows the models' comparisons based on detailed sentiment performances.

Finally, table 13 shows the overall performances of list of algorithms applied on three datasets. Among all these metrics the proposed approach DSA performance is more compare with existing approaches.

VIII. CONCLUSION AND FUTURE WORK

Sentiment analysis has many practical applications, such as understanding customer feedback, analyzing social media trends, and predicting the success of marketing campaigns. The performance of the proposed model improved by adding various models, including decision-based RNN and advanced BERT models. Decision-based RNNs are recurrent neural networks with a decision layer to make predictions based on the previously hidden states. This model is effective in sentiment analysis tasks because it can capture the temporal dependencies between words and the context in which they appear by using ABSA and PBSA.

On the other hand, the Advanced BERT model was used as a pre-trained transformer architecture to learn contextual representations of words and phrases. A fine-tuned SGD model was used to improve the training for the BERT model, and this method has performed admirably in various NLP tasks, including sentiment analysis. Researchers have proposed combining decision-based RNNs with advanced BERT models to enhance the accuracy of sentiment analysis further. This hybrid approach leverages the strengths of both models by using the decision-based RNN to capture temporal dependencies and the advanced BERT model to learn contextual representations of words and phrases. Combining decision-based RNNs and advanced BERT models offered outcomes for sentiment analysis tasks, obtaining outstanding performance on various benchmark datasets. However, it is important to understand this hybrid approach's capabilities and limitations and explore its potential applications in other natural language processing tasks. In the future, advanced deep learning models will be used to extract the sentiment analysis on text data, images, and videos. Extraction of sentiments from various images and videos is a difficult task. Thus, obtaining highly accurate sentiments requires a high-potential technique.

REFERENCES

[1] R. Liu, Y. Shi, C. Ji, and M. Jia, "A survey of sentiment analysis based on transfer learning," *IEEE Access*, vol. 7, pp. 85401–85412, 2019.

[2] A. Krishna, V. Akhilesh, A. Aich, and C. Hegde, "Sentiment analysis of restaurant reviews using machine learning techniques," in *Emerging Research in Electronics, Computer Science and Technology* (Lecture Notes in Electrical Engineering), vol. 545, 2019, pp. 687–696.

[3] K. Ayyub, S. Iqbal, E. U. Munir, M. W. Nisar, and M. Abbasi, "Exploring diverse features for sentiment quantification using machine learning algorithms," *IEEE Access*, vol. 8, pp. 142819–142831, 2020.

[4] Y. Bengio, Y. LeCun, and G. E. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, pp. 58–65, Jun. 2021.

[5] A. Viloria, N. Varela, J. Vargas, and O. B. P. Lezama, "Comparative analysis between different automatic learning environments for sentiment analysis," in *Proc. Int. Symp. Distrib. Comput. Artif. Intell.* Cham, Switzerland: Springer, 2021, pp. 134–141.

[6] F. Iqbal, J. M. Hashmi, B. C. M. Fung, R. Batool, A. M. Khattak, S. Aleem, and P. C. K. Hung, "A hybrid framework for sentiment analysis using genetic algorithm based feature reduction," *IEEE Access*, vol. 7, pp. 14637–14652, 2019.

[7] H. T. Rauf, U. Shoaib, M. I. Lali, M. Alhaisoni, M. N. Irfan, and M. A. Khan, "Particle swarm optimization with probability sequence for global optimization," *IEEE Access*, vol. 8, pp. 110535–110549, 2020, doi: 10.1109/ACCESS.2020.3002725.

[8] J. Odili, M. N. M. Kahar, A. Noraziah, and S. F. Kamarulzaman, "A comparative evaluation of swarm intelligence techniques for solving combinatorial optimization problems," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 3, May 2017, Art. no. 172988141770596, doi: 10.1177/1729881417705969.

[9] P. Stodola, K. Michenka, J. Nohel, and M. Rybanský, "Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem," *Entropy*, vol. 22, no. 8, p. 884, Aug. 2020, doi: 10.3390/e22080884.

[10] S. A. Ajagbe and M. O. Adigun, "Deep learning techniques for detection and prediction of pandemic diseases: A systematic literature review," *Multimedia Tools Appl.*, vol. 2023, pp. 1–35, May 2023, doi: 10.1007/s11042-023-15805-z.

[11] S. A. Ajagbe, K. A. Amuda, M. A. Oladipupo, O. F. Afe, and K. I. Okesola, "Multi-classification of Alzheimer disease on magnetic resonance images (MRI) using deep convolutional neural network (DCNN) approaches," *Int. J. Adv. Comput. Res.*, vol. 11, no. 53, pp. 51–60, Mar. 2021, doi: 10.19101/IJACR.2021.1152001.

[12] T. Gu, G. Xu, and J. Luo, "Sentiment analysis via deep multichannel neural networks with variational information bottleneck," *IEEE Access*, vol. 8, pp. 121014–121021, 2020, doi: 10.1109/ACCESS.2020.3006569.

[13] M. K. Hayat, A. Daud, A. A. Alshdadi, A. Banjar, R. A. Abbasi, Y. Bao, and H. Dawood, "Towards deep learning prospects: Insights for social media analytics," *IEEE Access*, vol. 7, pp. 36958–36979, 2019, doi: 10.1109/ACCESS.2019.2905101.

[14] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020, doi: 10.1109/TCSS.2020.3033302.

[15] P. Gupta, S. Kumar, R. R. Suman, and V. Kumar, "Sentiment analysis of lockdown in India during COVID-19: A case study on Twitter," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 992–1002, Aug. 2021, doi: 10.1109/TCSS.2020.3042446.

[16] A. Elouardighi, M. Maghfour, H. Hammia, and F.-Z. Aazi, "A machine learning approach for sentiment analysis in the standard or dialectal Arabic Facebook comments," in *Proc. 3rd Int. Conf. Cloud Comput. Technol. Appl. (CloudTech)*, Oct. 2017, pp. 1–8, doi: 10.1109/CloudTech.2017.8284706.

[17] P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal, and P. Muzumdar, "Automated classification of societal sentiments on Twitter with machine learning," *IEEE Trans. Technol. Soc.*, vol. 3, no. 2, pp. 100–110, Jun. 2022, doi: 10.1109/TTS.2021.3108963.

[18] R. Khan, "Social media analysis with AI: Sentiment analysis techniques for the analysis of Twitter COVID-19 data," *Crit. Rev.*, vol. 7, no. 9, pp. 2761–2774, 2020.

[19] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets," *IEEE Access*, vol. 8, pp. 181074–181090, 2020.

[20] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, "A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 114006, doi: 10.1016/j.eswa.2020.114006.

[21] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 667–676, Jan. 2018.

[22] S. Kumar, K. De, and P. P. Roy, "Movie recommendation system using sentiment analysis from microblogging data," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 4, pp. 915–923, Aug. 2020, doi: 10.1109/TCSS.2020.2993585.

[23] S. Bhatia, "A comparative study of opinion summarization techniques," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 1, pp. 110–117, Feb. 2021, doi: 10.1109/TCSS.2020.3033810.

[24] S. Davis and N. Tabrizi, "Customer review analysis: A systematic review," in *Proc. IEEE/ACIS 6th Int. Conf. Big Data, Cloud Comput., Data Sci. (BCD)*, Sep. 2021, pp. 91–97, doi: 10.1109/BCD51206.2021.9581965.

[25] M. A. Tayal, M. M. Raghuwanshi, and L. G. Malik, "ATSSC: Development of an approach based on soft computing for text summarization," *Comput. Speech Lang.*, vol. 41, pp. 214–235, Jan. 2017.

[26] E. Aslanian, M. Radmanesh, and M. Jalili, "Hybrid recommender systems based on content feature relationship," *IEEE Trans. Ind. Informat.*, early access, Nov. 21, 2016, doi: 10.1109/TII.2016.2631138.

[27] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[28] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," *Int. J. Comput. Commun. Control*, vol. 13, no. 1, pp. 50–61, 2018.

[29] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Syst. Appl.*, vol. 69, pp. 214–224, Mar. 2017, doi: 10.1016/j.eswa.2016.10.043.

[30] S. Tan, Y. Li, H. Sun, Z. Guan, X. Yan, J. Bu, C. Chen, and X. He, "Interpreting the public sentiment variations on Twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1158–1170, May 2014.

[31] Z. Jianqiang and G. Xiaolin, "Comparison research on text preprocessing methods on Twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[32] D. Tang, B. Qin, F. Wei, L. Dong, T. Liu, and M. Zhou, "A joint segmentation and classification framework for sentence level sentiment classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 11, pp. 1750–1761, Nov. 2015.

[33] M. K. Sohrabi and F. Hemmatian, "An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: A Twitter case study," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24863–24882, Sep. 2019, doi: 10.1007/s11042-019-7586-4.

[34] A. A. Aziz and A. Starkey, "Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches," *IEEE Access*, vol. 8, pp. 17722–17733, 2020, doi: 10.1109/ACCESS.2019.2958702.

[35] A. Deniz, M. Angin, and P. Angin, "Evolutionary multiobjective feature selection for sentiment analysis," *IEEE Access*, vol. 9, pp. 142982–142996, 2021, doi: 10.1109/ACCESS.2021.3118961.

[36] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020.

[37] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, 2020.

[38] J. Du, L. Gui, Y. He, R. Xu, and X. Wang, "Convolution-based neural attention with applications to sentiment classification," *IEEE Access*, vol. 7, pp. 27983–27992, 2019.

[39] F. Yin, Y. Wang, J. Liu, and L. Lin, "The construction of sentiment lexicon based on context-dependent part-of-speech chunks for semantic disambiguation," *IEEE Access*, vol. 8, pp. 63359–63367, 2020.

[40] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, and S. Seth, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, pp. 26172–26189, 2020.

[41] A. Rasool, R. Tao, M. Kamyab, and S. Hayat, "GAWA—A feature selection method for hybrid sentiment classification," *IEEE Access*, vol. 8, pp. 191850–191861, 2020.

[42] T. Bhattacharyya, B. Chatterjee, P. K. Singh, J. H. Yoon, Z. W. Geem, and R. Sarkar, "Mayfly in harmony: A new hybrid meta-heuristic feature selection algorithm," *IEEE Access*, vol. 8, pp. 195929–195945, 2020.

[43] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 874–888, Feb. 2021.

[44] K. P. Vidyashree and A. B. Rajendra, "An improvised sentiment analysis model on Twitter data using stochastic gradient descent (SGD) optimization algorithm in stochastic gate neural network (SGNN)," *Social Netw. Comput. Sci.*, vol. 4, no. 2, pp. 1–12, Feb. 2023, doi: 10.1007/s42979-022-01607-x.

[45] G. Zhao, Y. Luo, Q. Chen, and X. Qian, "Aspect-based sentiment analysis via multitask learning for online reviews," *Knowl.-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110326, doi: 10.1016/j.knosys.2023.110326.

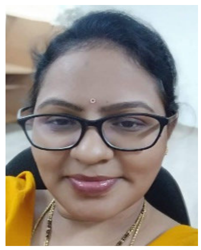
[46] H. Zhang, Y. N. Cheah, O. M. Alyasiri, and J. An, "A survey on aspect-based sentiment quadruple extraction with implicit aspects and opinions," Tech. Rep., Jun. 2023, doi: 10.21203/rs.3.rs-3098487/v1.

[47] O. M. Al-Janabi, M. K. Ibrahim, A. Kanaan-Jebna, O. M. Alyasiri, and H. J. Aleqabie, "An improved bi-LSTM performance using Dt-WE for implicit aspect extraction," in *Proc. Int. Conf. Data Sci. Intell. Comput. (ICDSIC)*, Nov. 2022, pp. 14–19, doi: 10.1109/ICDSIC56987.2022.10076109.

[48] K. V. Deshmukh and S. S. Shiravale, "Priority based sentiment analysis for quick response to citizen complaints," in *Proc. 3rd Int. Conf. Converg. Technol. (I2CT)*, Apr. 2018, pp. 1–5, doi: 10.1109/I2CT.2018.8529722.

[49] S. Hussein, "Twitter sentiments dataset," Tech. Rep., 2021, doi: 10.17632/z9zw7nt5h2.1.

[50] H. Cai, R. Xia, and J. Yu, "Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 340–350.



PUTTA DURGA received the dual bachelor's degree from the Chadalawada Ramanamma Engineering College, Tirupati, and JNTUA, Andhra Pradesh, India, and the dual master's degree in computer science and engineering from the SRK Institute of Technology, Vijayawada, and JNTUK. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh. She is having four years of experience in teaching. Her current research interests include NLP, deep learning, digital image processing, and machine learning.

DEEPTHI GODAVARTHI received the Ph.D. degree from Andhra University, Andhra Pradesh, India, in 2022. She is currently an Assistant Professor with the School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh. Her research interests include AI, NLP, computer vision, and block chain technologies.