SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Novel Method for Sentiment Analysis in Social Media Data Using Hybrid Deep Learning Model

Harika vanam[1,*], Jeberson Retna Raj[2]

[1] Computer Science Department, Satyabhama Institute of Science and Technology Chennai, India
[2] Department of Information Technology, Satyabhama Institute of Science and Technology, Chennai, India

ARTICLE INFO

Article history:
Received 22 march 2023
Received in revised form 8 June 2030
Accepted 28 August 2023
Available online 3 September 2023

Keywords:
Deep Learning; Social media; Autoencoder; Sentiment analysis; Feature extraction

ABSTRACT

It is common practise to employ a contextual mining approach called sentiment analysis (SA) to glean subjective but potentially helpful information from textual data. In order to recognise, analyse, and extract answers, states, or emotions from the data, it employs Natural Language Processing (NLP), language understanding, forensics, and cognitive science. Significant progress towards a better SA model may be made using the features analysis method. In recent years, feature extractions have made extensive use of GloVe and Word2vec embedding models. They require a huge corpora of text data for training and creating accurate vectors, but they ignore emotional and contextual information in the text. Out-of-Vocabulary (OOV) words are not considered while creating these vectors, hence some information may be lost if these methods are used. The limited availability of annotated data is another obstacle to sentiment categorization. Misclassification may occur when there is a discrepancy between the review and the label. In this research, we offer a hybrid SA model capable of overcoming the challenges posed by missing ratings and reviews due to noise, out-of-context phrases, and context. To investigate and conduct sentiment and appropriate analysis, this study proposes an Autoencoder Bi-directional Recurrent Neural Network (ABRNN) based on Bi-directional Encoding from Transformers (BET). The reviews are initially categorised by their polarity ratings using the zero-shot categorization. Then, the data is fed into a pre-trained BET system to extract embeddings based on the semantics and context of sentences. The neural network, made up of expanded and hybrid LSTM, was then fed the acquired contextual embedded vectors. For extracting both global and local spatial contextual characteristics from the embedded data, this approach employs expanded approach in place of traditional approach. The complete phrase sequencing is performed with the help of Hybrid Long Short-Term Memory (HLSTM). Measures of accuracy, precision, recall, f1-score, and area under the curve (AUC) are used to assess the ABRNN model on four separate text datasets from different domains. Because of this, ABRNN may be utilised well for SA processes on social media evaluations, with no loss of data.

* *Corresponding author.*
E-mail address: hv.it@kitsw.ac.in

https://doi.org/10.37934/araset.32.1.272289

1. Introduction

Even something as straightforward as customer feedback or a discussion forum might contribute to new intelligence goals made possible by the growth of information technology through human-centric meaning [1]. The rapid expansion of such huge datasets produces a plethora of anecdotal evidence. Natural language processing (NLP), data analysis, and knowledge discovery have all made significant contributions to our understanding of sentiment analysis. It is a technique used to display and track how well information about public opinion and consumer feedback is understood across a wide range of entities and characteristics, including online communities, companies, and a wide variety of goods and services. People's feelings and thoughts about things and characteristics that have either good or negative connotations are also assessed [2].

Numerous features, including n-gram features, have been explored in sentiment analysis across a number of fields. Scientists have traditionally studied sentiment analysis at three levels of subdivision, as shown in their significant result on document-based sentiment analysis (DOCSA): the document level, the phrase level, and the aspect level. Sentiment analysis is often viewed as a method of document classification when a more in-depth look is taken [3]. If we were to classify documents in this way, the description of each document would play a pivotal role, since it would focus on the most important data sent by strings. We want to do sentiment analysis on complete documents as part of our investigation. With the rise of the Internet and other forms of digital communication, the ease with which people from all over the world can communicate with one another and exchange ideas, feedback, and feelings has increased dramatically. Overview of Sentiment Analysis Process is shown in Figure 1.

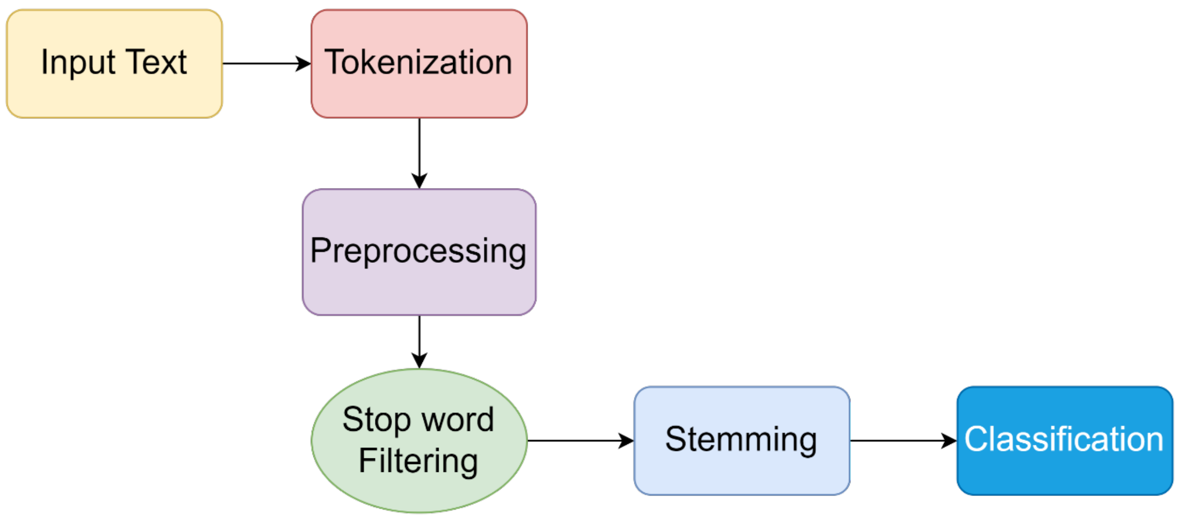

Fig. 1. Overview of Sentiment Analysis Process

The use of social media sites such as Facebook, Twitter, LinkedIn, Tumblr, YouTube, etc. has increased in recent years. These sites are used by businesses, individuals, and governments to do business, advertise goods and services, have conversations about relevant issues, and start campaigns and raise awareness. Businesses now have more opportunities because to the widespread availability and development of social media platforms [4]. They employed several models for picking up on people's reactions and perspectives. Several methods have been used to analyse social media content for intelligence gathering, monitoring unethical behaviour in the workplace, and sentiment analysis of consumer feedback.

SA, also referred as sentiment detection, is an important part of natural language processing (NLP) that aims to assist users in analysing and identifying the emotions present in subjective writings. It is widely employed for determining the polarity of textual data in social media such as blogspot, Facebook comments, review sites, etc. The polarity of an item's sentiment is the degree to which the stated feelings of a user are positive, negative, or neutral in relation to the object in question. Identifying the overarching tone of several product evaluations posted on the web can help consumers make well-informed purchases. SA is a subset of text categorization that, depending on the need, can be broken down [5].

While classifying at the document level, the entire article is treated as a single unit, but when classifying at the sentence level, a sentence can be thought of as a mini document. On direct contrast, aspect-based SA zeroes in on a single aspect or phrase and its related polarity [6]. Since SA at the sentence level contains both the partiality and impartiality of the sentence, it is a promising new area in textual data for feature learning. Similarly, SA entails essentially four key steps: preprocessing, feature extraction, categorization, and analysis of results across many domains such as film reviews, political opinion forecast, flight customer feedback, Amazon.com reviews, and so on.

Feature extraction is a crucial part of the aforementioned processes since it increases the accuracy of the classification. Feature extraction strategies may be broken down into lexicon-based strategies and machine learning/deep learning-based strategies. The model in machine learning-based approaches looks for patterns in the data, whereas in lexicon-based approaches it is given lists of negative and positive terms to use [7]. These word counts reflect the total number of words used in each sentence. How often words that are optimistic or pessimistic appear in a text determines its tone. As a result, the lack of generalizability in these lexicon-based approaches renders them less useful in fields where specialised vocabularies are not available. However, due to a lack of robust language resources, these methods often have a poor accuracy rate. Techniques for machine learning are further divided into two categories: supervised and unsupervised [8].

In order to determine how well a model performs, supervised methods first need to have it trained on labelled data. Contrarily, unsupervised methods do not rely on human labels or categories to train the model, allowing it to operate autonomously. Bag-of-Words (BoW) and N-grams are two of the most popular methods used in machine learning for feature extraction. Count Vectorizer (CV) and Term Frequency-Inverse Document Frequency (TF-IDF) are two of the most used methods for extracting features from data (TF–IDF). These methods rely on a single "hot" representation of the document, where the vocabulary size is proportional to the number of visible words [9]. This results in a high-dimensional feature space and increases the difficulty of scaling. Given its foundation in the BoW model, TF-IDF is limited in its ability to catch syntactic and semantic nuances within a sentence's word sequences.

Effective extraction of feature representation, however, is dependent on both the choice of relevant features and the chosen approach. Word-embedding models were presented to deal with the shortcomings of the previously mentioned feature extraction strategies. Word-embedding models get answers by deducing grammatical and semantic relationships between words. Word embedding's popularity has shifted the emphasis of many studies back to neural networks [10]. The difficulty of sentence-level categorization has been widely studied since it is representative of a more general issue with SA. For text mining, the word-embedding models Word2vec and Glove are among the most popular choices.

The CBoW makes its predictions by looking at the surrounding text, while the Skip-gram relies on the centre word or target word to make its predictions. In contrast, this approach uses an approach, which in turn creates vectors using the matrix factorization technique and word co-occurrence. For SA applications, the standard model emphasis on the selected features is inadequate. These incredibly efficient methods, however, do have a few drawbacks that need fixing. In order to train and generate embeddings for each word, models like Word2vec and GloVe need a massive corpus. These embedding approaches can only handle words that are already in their vocabularies, as they build feature vectors for those words and ignore any that are not.

Similar words in various phrases may have comparable vector representations, which is another drawback of these approaches. They also noted that same words in different phrases could have a distinctive meaning depending on the surrounding material. Moreover, in such methods, words with opposing emotive weights, like "bad" and "good," could have the nearest trajectories. There might be emotional and intellectual detachment as a result. The over 200 million tweets processed by Twitter every day make it one of the most popular social media platforms. Because of the limited character space in a tweet, typos are all too often. Mistakes in spelling and other forms of linguistic inconsistency that one encounters on social media can be challenging at times. Furthermore, the difficulty of the categorization assignment is exacerbated by the dearth of publicly accessible annotated data. A model that can zero down on semantic and syntactical aspects and extract emotive and contextual information without being constrained by domain dependency needs to be proposed.

To improve the quality of sentence-level SA, it is recommended to use an Autoencoder Bi-directional Recurrent Neural Network (ABRNN) model to address the issues mentioned above. The major contributions of the research work are as follows:

- The zero-shot approach is first utilised for data annotation, and then the BET approach is applied to produce controlled model. Extracting these emotive characteristics over these features requires a dilated CNN model with varying dilation rates.
- Learning long-term connections both forwards and backwards among word sequences in a long message is used by the HLSTM model.
- A variable selection CV technique was used to fine-tune the suggested model's parameters.
- The effectiveness of the proposed BET-based CBRNN model is evaluated via a comparison to existing methods.

2. Literature Survey

A key step in text mining or SA, feature extraction is also highly sensitive to the techniques employed. More recently, deep learning algorithms have been applied in the domain of SA to learn word embeddings. The goal of word embeddings is to record linguistic relationships and similarities between words. Word2vec and glove embeddings are two popular examples of unsupervised word embeddings [11]. Words in a comparable context are assumed to possess the same meaning in these methods, hence they generate highly contextualised features or vectors.

The main issue with this assumption, however, is that it leads to a similar vector being created for a number of semantically unrelated words that appear often together in a small area. Typically, these methods map contrasting phrases onto adjacent vectors, despite the fact that the words' true meanings are diametrically opposed to one another. For SA in a variety of settings, a unique neural word embedding-based strategy was proposed [12]. They fixed one of the biggest problems with previous approaches, which was how poorly they fared when applied to domains other than the one in which they were initially taught. Their new method was more effective than the previous one. These methods, however, needed a sizable training corpus to generate reliable vectors.

A new vector for word embeddings was proposed this year. Model Improved Word Vector (IWV) was a fusion of lexicon-based techniques, a POS tagger, word2vec/glove models, and a word localisation approach. Even while retraining this embedding model on four GPUs resulted in slight accuracy gains, the required computing power was substantial [13]. The same may be said of the suggested word2sent emotional embedding paradigm. Using a combination of the CBoW and senti-wordnet-lexicon models, embeddings for each word were discovered using their immediate neighbours as input. Syntactic and semantic aspects were preserved, and emotions were picked up on in an indirect way.

Therefore, a CNN classifier was utilised to categorise sentences from four different datasets. The fundamental drawback of these approaches is the time and effort required to determine the opinion/emotion disposition of each and every word in the dictionary. It's also feasible that the emotional/opinion orientation terms change depending on the context. The sentiment information in conventional embedding systems used mostly for sentiment analysis may be lost if OOV terms are employed [14]. One potential downside of such methods is that they may lump together phrases with identical wording. But it seems to reason that words in various phrases would have varying contexts. For the past two years, many text classification tasks have relied on vectors generated by transformer-based word embedding models.

Additionally, a BERT training algorithm was employed to improve the efficiency evaluations. The authors have created a method for identifying people's personalities using BERT embeddings. They found that the BERT model may greatly improve the accuracy of character identification from text [15]. The authors have examined many deep models for SA of medication reviews, each utilising a unique embedding. They used LSTM embeddings for clinical BERT that had already been pre-trained, however the results were subpar. After comparing them, researchers found they combined with CNN yielded the best results in terms of accuracy.

Overall, the findings of the aforementioned studies suggest that the shortcomings of traditional word embedding models might be corrected by employing a transformer-based strategy. Over the course of the past several years, deep learning models have been utilised in sentence-level SA in a number of different domains in order to circumvent the limitations of more traditional machine learning models [16]. CNN, while helpful for little amounts of textual material, may not be up to snuff for lengthy assessments. But LSTM can handle textual material that is quite lengthy.

However, this may be difficult for data with extensive time dependencies. The use of Bi-LSTM has allowed for the management of extremely lengthy sentence dependencies. Because the deep learning models had performed better when used together than separately, some research had suggested using hybrid models for SA. The advantages of convolutional neural networks (CNNs) and long short-term memories (LSTMs) have been used to discern the positive and negative polarity of drug evaluations. Each evaluation indicated the medication user's degree of pleasure with a rating between 1 and 5. For more nuanced polarity detection in IMDB and Amazon reviews, researchers have deployed a convolutional neural network (CNN) and long short-term memory (LSTM) model embedded in the Word2vec framework.

On both datasets, the CNN-LSTM hybrid model attained 91% accuracy. Another study developed a bidirectional Convolutional RNN model. When applied to the SA of IMDB dataset, a Bidirectional Gated Recurrent Unit (Bi-GRU) is used to link together two layers of a Convolutional Neural Network (CNN). While CNN retrieved a large corpus of sentence-level features, Bi-GRU used long-term reliance to acquire chronological features. These models excelled solely in their designated domain. As a result, they suggested a hybrid model built on Convolutional LSTMs (CO-LSTMs) that could be easily scaled to handle data from a variety of domains. In order to sequentially analyse the lengthy text, LSTM was employed, and a deep convolution network was used to extract relevant features via a pooling layer.

Although these models are able to process sequences of arbitrary length, incorporating them into the extracting features phase of a deep neural network raises the complexity of the feature space.

These models also suffer from the flaw of treating all features equally. The researchers have suggested a model that combines Bi-LSTM and a self-attention technique to fill up the gaps left by earlier research [17]. The model used many routes to collect relevant information for text content categorization.

For processing both lengthy and brief passages, the authors presented a convolutional neural network (CNN) and attention-based model. When it came to extracting the features' historical and prospective contexts, two separate layers were used. After that, we focused on quantifying the value of various nouns. They have been widely utilised for targeted sentiment categorization, although prior work showed they still had certain drawbacks [18]. As the authors have previously noted, there are two fundamental issues with social media SA because of the size restrictions imposed by conventional CNNs. This problem was that CNN couldn't handle capturing long-term dependence patterns when it came to semantics.

The increasing complexity of convolutional kernels has led to a dramatic increase in the total number of parameters [19]. Dilated CNN (D-CNN) is a newer kind of CNN that has been presented as a solution to the issues that plague the traditional form. As a result, we can conclude that while traditional methods are straightforward to understand, require little in the way of hardware, and work well with small-size datasets, they have difficulty with complex classification problems and necessitate expert knowledge when it comes to constructing sentiment lexicons [20]. There are some shortcomings of traditional approaches that may be addressed by using approaches, such as reducing the dependence on physical characteristics.

Despite this, since SA is somewhat comparable to consecutive design, and there is a danger that contextual and emotional information will be lost if CNN and RNN systems are combined [21]. Moreover, the model may get more complicated since it must use many convolution kernels to learn the high-level relevant data. Therefore, this research introduces a convolutional neural network (CNN) expansion and a long short-term memory (LSTM) based categorization model for SA of textual content in both long and short evaluations [22, 23].

3. The Proposed System

Word embedding, commonly known as the word representation approach, is a method for encoding words as numbers. Since machine learning techniques are ineffective when applied to text data has a crucial approach. This is a vocabulary-based approach that changes individual words into numerical representations. To improve its performance, a neural network can be trained on a large body of text. There is a wide variety of embedding methods. The Hierarchy of the operations in the proposed system are as follows: Data Cleaning, Tokenization, Pooling, Model Training, Sentiment extraction. In the proposed model, we are using three CNN layers named as a convolutional layer, a pooling layer, and a fully connected layer.

To further understand these methods, we may categorise them into two groups: frequency-based and prediction-based embeddings. Frequency-based encoding techniques, such as TF-IDF, co-occurrence matrices, and CV, produce text vectors by tallying up the occurrences of commonly used words. In contrast, the prediction-based embedding approaches use prior information and a neural network to vectorize a word. The skip-gram and CBOW models are two popular implementations of this technique. Tokenization is a method that breaks down larger chunks of text into smaller, more manageable units called tokens.

WordPiece tokenizer, one of several types of tokenizers described here, builds a vocabulary by first extracting all the symbols, subwords, and words from the training data. Complete words, Phrases, Prepositions, and Adverbs make up the first three categories of the vocabulary list. Word stems, individual characters, non-starter subwords that are prefixed by ## to indicate this instance (such as "se" in "search," which is assigned the same matrix as the standalone succession of letters "se" in "go obtain se"), and individual words.

In this method, the tokenizer checks to see if the term is already in the dictionary before attempting to tokenize it. If it doesn't work, it tries to divide the word into as many parts as the vocabulary allows, and if that fails it resorts to character-by-character breakdown. The next step is tokenization, which uses the established vocabulary. In order to build the BERT tokenizer, the aforementioned WordPiece algorithm is used. As a result of these processes, the BERT tokenizer provides token identifiers and focus masks. The BERT model will use the tokenizer's output as an input in order to produce context - specific vector representation.

An innovative approach in machine learning, transfer learning repurposes learned skills to solve problems in unrelated domains. Specifically, the aforementioned BERT model was presented by Google as a new kind of transformer in 2018. Because, the BERT model of language is progressive. For each element of the input sequence, $Y = (m_0, m_1 \dots m_n)$, BERT generates a contextualised vector representation $I = (i_0, i_1 \dots i_n)$. Due to its flexibility as a language representation system, it is possible to complete its work by means of an encoder. In order to generate encoded representations of text, encoders borrow a neural network design from transformers. There are four levels of encoders in the relevant BERT-Mini. In an encoder unit, the multi-head attention and feed-forward layers serve as the two sub-layers.

As is evident from Figure 2, each encoder layer is composed of two distinct procedures. The first kind is a multiple recovery layer, which uses a number of procedures to manipulate metrics.

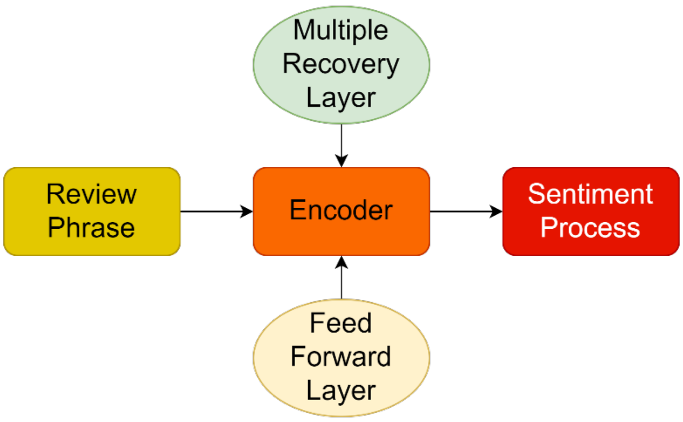

Fig. 2. Encoder process layer

Each "head" of the multi-head attention is analogous to a separate form of self-attention, and together they form the overall multi-head attention. The self-attention pinpoints the connection between each word in a sentence. Defining the process of self-attention is crucial for full comprehension. The self-attention process is depicted graphically in Figure 3.

$$a = softmax\left(\frac{RL^T}{\sqrt{e_l}}\right) w \tag{1}$$

where R represents a query vector, L represents a key vector and w represents the vector consisting of values. The l-dimension has been denoted by e_l. Dot product (QKT) of query and key matrices is required for similarity score computation. After that, we use $\sqrt{e_l}$ to partition the key matrix (RL^T).

The score matrix is then normalised and obtained through the softmax method. When the scoring matrix is multiplied by the vector w, we get the attention matrix a.

Furthermore, single attention-head averaging precludes the model from attending to distinct representations and subspaces in different places, but multi-headed attention permits this. The feed-forward consists of two distinct linear combinations that are connected through a ReLU activation represented by n_b.

$$N_b(R, L, W) = concat(a_0, a_1 \dots a_i) \tag{2}$$

The method is used in each location independently and in the same way. The mathematical expression for this is:

$$G_o = \max(0, yX_1 + c_1)\, X_2 + c_2 \tag{3}$$

Common components of a CNN include convolution and sub-sampling methods, both of which can be used over several layers. In NLP, CNNs are used to glean contextual information. Convolutional, pooling, and fully linked layers make up CNN. When using a convolution layer, it is common practise to apply it to input data features. When implemented to textual information, it aids in the feature extraction from a word or sentence level description. The feature-map produced by the convolution layer is made more compact by the pooling layer. It is a powerful method for minimising the number of training parameters required from high-dimensional input information.

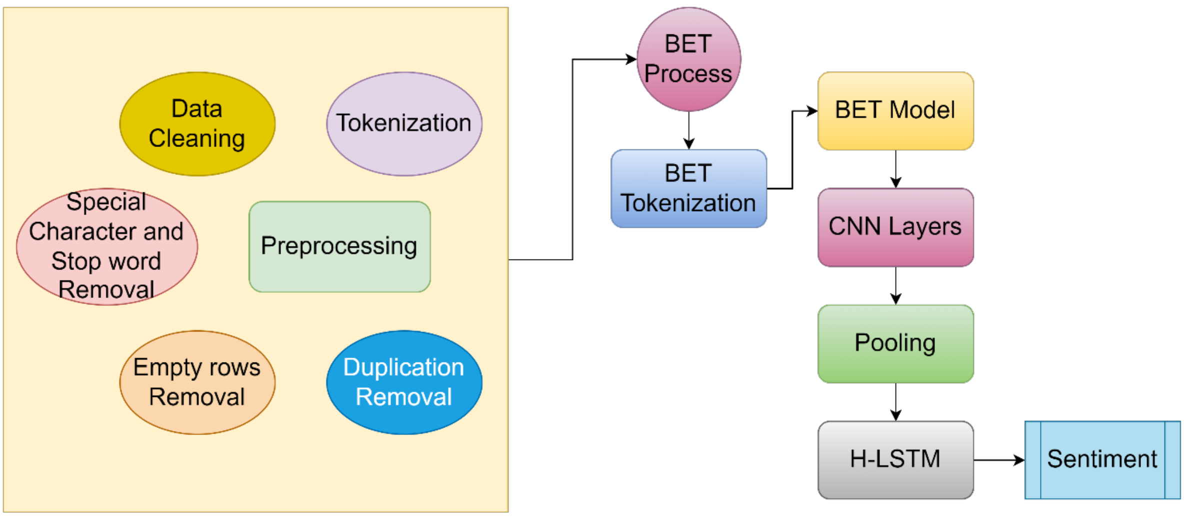

Fig. 3. Proposed System Architecture

There are several types of pooling processes, including maximum pooling, average pooling, total pooling, etc. Proposed system architecture is depicted in Figure 3. Max pooling returns the single most important feature of an area, such that normal merging returns the average of all patch features. An aggregated feature vector is produced when the pooling layer has finished its work. Once a fully connected layer is reached, the acquired vector may be sent there. Although not essential, pooling can expand the receptive field of the convolution kernel in traditional CNN. Contrarily, there is often a substantial amount of data loss when pooling processes are used excessively.

Data tokenization and padding, data transformation, and the BERT model's identification of contextual embeddings are all part of the work suggested here. Annotating data and extracting embeddings from input data are two of the main goals of this study, both of which have been accomplished with the help of the BERT model. In order to annotate the intensities, the zeroshot classification approach was utilised once the data was cleaned. After embeddings are retrieved, they are employed by the BERT model to generate feature vectors, which are then fed through three enlarged convolution layers to produce a feature map.

Next, a universal max-pooling procedure is performed on the merged layers in order to choose the most important features from the resulting feature map. To learn the sequential relationships between the combined features, Bi-LSTM is employed. To avoid over-fitting, a 0.2-point dropout has been included in this layer. The Bi-LSTM output is sent to the hidden layers, and from there it is predicted by the fully connected or exponential layer. In the sigmoid layer, the loss function is the ubiquitous bidirectional cross-entropy.

Phase 1: Pre-processing

There is typically no structure to the material in assessments of social media sites. With the help of the feature representation methods, the unstructured data is transformed into a more manageable format. It is essential to clean up the input before performing any text categorization task, as the underlying dataset shows many irregularities that might mislead the embedding system throughout the tokenization phase. The casual nature of the language employed in social media increases the likelihood of noise in the data collected from these sources. There might be typos, accented letters, hyperlinks, symbols, etc.

Cleaning data and extracting the useful bits using regex and text hero libraries. Text mining techniques are case-sensitive, so afterward, we change the case of all the reviews to lowercase. In order to discover a connection between terms and to display data, exploratory data analysis (EDA) is conducted. The NLTK library has been used to conduct text normalisation procedures like lemmatization. Reducing the quantity of phrases that are often used in nature can be accomplished by a method called lemmatization, which involves reducing an alternate version of a word by employing its fundamental form, or lemma Even though "go," "going," and "gone" are all technically different words, in this context, "go" will function as the morphological form for all three.

After finishing the data cleaning process, we use the provided reviews to generate labels with zero-shot-BERT. Use of the BERT scale to classify evaluations as good or unfavourable is widespread. In addition to determining whether a statement is good or negative, zero-shot also reveals data on the strength of the feeling expressed. It's rather quick and can be utilised for delivering information over the internet requiring a major speed-to-performance trade-off. After extracting the strengths, such as positive, negative, and neutral ratings for every review, the data in this study is then categorised according to a set of rules. When the review's negative score exceeds the sum of its neutral and positive values, we classify it as negative. A class is considered favourable if it has a higher total score than the other two groups combined. In the end, this procedure yielded fully pre-processed data that may be utilised to generate a numerical representation of these critiques of text.

Phase 2: Token Separation

Following data is cleaned and categorized, the next step is to vectorize the words and submit them for classification. This process is commonly referred to as "tokenization." In order to construct input ids and attention masks for subsequent processing, we first tokenize our text with the WordPiece tokenizer. As a means of making sure each review is the same length, padding and truncation are required. Based on the progressive probability density, the longest tweets can be 25 characters long, whereas the longest reviews can be 250 characters long. Therefore, we established a maximum length for tokens based on the length of the text they would represent.

The identifiers of the words in our sentences are contained in the input identifiers, and an attention filter directs the model's emphasis to the phrase of interest. This is due to the fact that input sentences that are too short will be patched using an extra special character. To guarantee that our BERT system does not consider any irrelevant information while generating experiential embeddings for this symbol, we use a focus filter.

Phase 3: Bi-directional Encoding from Transformers (BET)

Tokenizer outputs ids and focus masks for use by the BET model. With BET, you can get contextualised word embeddings, which is a major advantage over Word2Vec models. In contrast to word2vec, which provides a static representation for each word regardless of the context in which it appears, Bert generates word representations that are constantly modified by the words surrounding them. There's also the argument that, from a parameter's perspective, training BET to perform a certain task is inefficient. However, a pre-trained BET model has been used due to the issue of computation expense. The Bert-Mini model was employed for the study; it has a dimension of 256, four encoder layers, and four attention heads. Each encoder's outputs are fed into the next, with the last encoder producing the phrase's contextual embeddings.

Phase 4: Bi-directional Convolutional layer

After applying three enlarged convolution layers to the input vectors, emotional and semantic aspects may be extracted from the data as the receptive field expands exponentially. We can extract the salient features of each phrase vector. Since ReLU has been shown to be six times quicker than tanh and activation function, it is employed to avoid the gradient vanishing problem. Each of the three convolution layers features 64 filters and a 3×3 kernel size, and the matching convolution layer's dilation rate is 1, 2, or 3. A low dilation approach, which zeroes in on certain words and phrases, has been utilised to establish lasting semantic traits. Once the feature maps for all three layers have been collected, they are concatenated into a single feature map using a holding process. While they are joined, a single feature map is generated. We can expect the output vector as:

$$Y = y_0, y_1 \dots y_n \tag{4}$$

Phase 5: Max-Pool Layer

After a m×n feature matrix is produced by the concatenation layer, a 2×2 filter is used to accomplish the maxpooling procedure. Max-pooling mode selects the largest value at each filter patch during a filter traversal. Therefore, the max-pooling layer should produce a convolutional feature map that only included the most salient and consequential elements from the previous feature space.

Phase 6: Hybrid LSTM Layer

The resulting feature vectors are then progressively analysed in both forward and backward directions by the HLSTM layer, which receives its input from the max-pooling layer. This layer made use of an LSTM with 256 units and a dropout of 0.1%. The below is an equation that may be used to describe the HLSTM:

$$g_{mHLSTM} = \left[g_m^{forward}, g_m^{backward}\right] \tag{5}$$

The HLSTM's output feature vector is then normalised before being fed towards the dense layer. The dense layer feeds its outputs into the fully-connected layer, which then uses them to forecast responses.

7. Feature Vector Layer

Introducing the activated sigmoid function to the input vectors that was acquired after extraction of features from the result of the previous deep network yields the frequency of dispersion for every classification. This can be represented as:

$$Prob_{\mu}(d^{k}) = \frac{f^{pk}}{1+f^{pk}} \tag{6}$$

where $Prob_{\mu}$ defines the probability distribution value for k^{th} value for the output vector obtained in the pk of the k^{th} value. The sigmoid layer's recovered probabilities are then used to compute sentiment heterogeneity by comparing the two sets of data. It's a measure of randomness that's computed with the help of the binary cross-entropy coefficient. The values are hence discretized as the range of $[0,1]$ which describes the positive and negative sentiment reviews. Also, the loss value can be calculated as:

$$loss = -\sum_{k=1}^{s} B(d_k) \times \log\left(Prob_{\mu}(d^{k})\right) \tag{7}$$

In this context, s stands for the total number of possible labels or classifications. We compute loss by contrasting actual quantities denoted by $B(d_k)$ with predicted values. This loss function is used primarily to close the gap between observed and anticipated values.

The first step is to represent the input text data in a format that can be processed by the CNN. This typically involves converting words or characters into numerical vectors. Common techniques for this include word embedding models like Word2Vec or GloVe, which capture semantic relationships between words. The convolutional layers are the core building blocks of a CNN. In the context of text analysis, the convolutional operation involves sliding a filter (also known as a kernel) of fixed size over the input data. This operation allows the model to extract local features or patterns from the text. For example, a filter of size 3 would capture three consecutive words or characters at a time. During the convolution operation, the filter is multiplied element-wise with the input data, and the resulting values are summed.

This process is repeated for different positions of the filter across the input, producing a feature map that represents the presence of specific patterns in the text. After the convolution operation, an activation function (such as ReLU or tanh) is applied element-wise to introduce non-linearity into the model. Non-linearity helps the model learn complex relationships between the input data and the corresponding sentiment. Subsequently, pooling layers are commonly used to reduce the dimensionality of the feature maps. Max pooling is a popular pooling operation where the maximum value within a region (e.g., 2x2) of the feature map is selected, discarding the other values. This downsampling helps in reducing the spatial dimensions while retaining the most salient features.

The output of the convolutional and pooling layers is flattened into a 1-dimensional vector. This vector is then passed through one or more fully connected layers, which are traditional neural network layers where each neuron is connected to every neuron in the previous layer. These layers allow the model to learn higher-level representations and capture more global dependencies in the data. The final layer of the CNN is typically a softmax layer, which provides a probability distribution over the possible sentiment classes (e.g., positive, negative, neutral). The model predicts the sentiment based on the highest probability in the output distribution.

The CNN is trained using labeled data, where the input text samples are paired with their corresponding sentiment labels. The training process involves minimizing a loss function, such as categorical cross-entropy, between the predicted sentiment probabilities and the true labels. This is typically done using backpropagation and gradient descent optimization algorithms. By adjusting the architecture, hyperparameters, and training data, CNNs can be tuned to achieve good performance in sentiment analysis tasks.

4. Results and Discussion

The scalability and efficiency of the suggested approach for correctly evaluating procedures on heterogeneous corpora of differing domains and sizes is tested on many datasets. Each of the four state-of-the-art datasets comes from a different industry. We have only used reviews that were either highly praised or critically panned in our studies. The numerous factors that were evaluated to ascertain the final result are detailed below. Table 1 lists the evaluation outcomes of the proposed model.

In grid search CV, the optimal values of hyperparameters are specified, and the suggested model is subsequently evaluated using those specifications. We then choose the optimum settings that give us the greatest outcomes.

Table 1
Evaluation Outcomes

	CNN	**LSTM**	**ABRNN(Proposed)**
Precision	0.91	0.89	0.98
Recall	0.92	0.88	0.97
F-Score	0.94	0.95	0.98
Accuracy	0.96	0.92	0.99
AUC	0.91	0.88	0.98

The proposed model's hyperparameters are learning rate, number of iterations, and density size. For the ABRNN model, the best values were found by using a grid search; they are as follows: 64 for CNN filters, 256 for BLSTM units, 0.1 for the learning rate, 64 for the batch size, and 32 for the dense size. This chapter provides a comprehensive review of the results of the experiments. Figure 4 depicts the performance comparison of ABRNN with existing systems.

In order to evaluate the efficacy of the suggested ABRNN method, we calculated its average recall, precision, f-score, and accuracy scores. It is the confusion matrix that is used to determine these ratings. Besides using the Area Under the Curve (AUC), the Receiver Operating Characteristic (ROC) is also employed to assess the model's performance. Performance matrices are used extensively in several text classification tasks, including SA. There is a comparison of the BERT-based ABRNN model to other deep learning models with various embedding strategies. A confusion matrix is a graphical display of the outcomes of a classification prediction for a certain situation. ROC Curve for the proposed approach is shown in Figure 5.

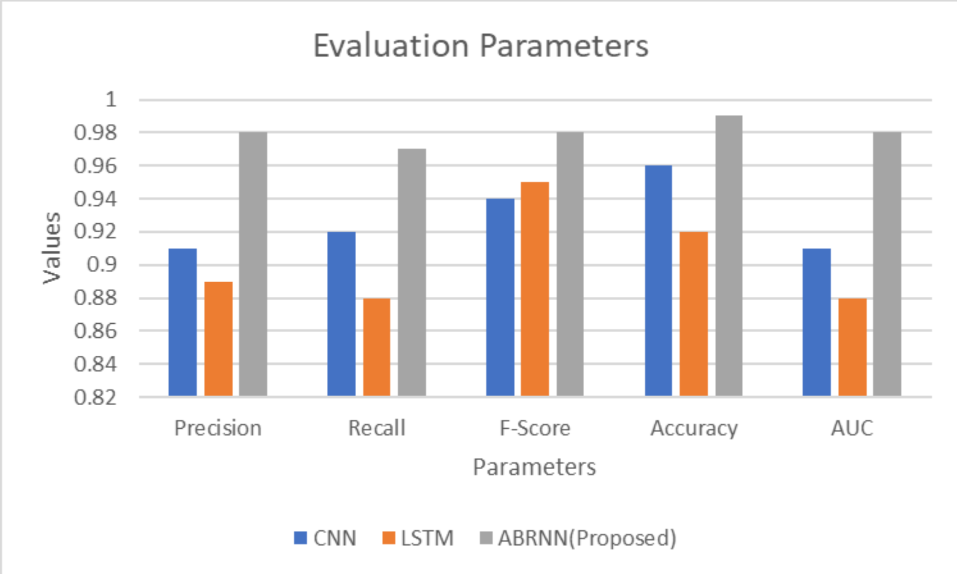

Fig. 4. Performance comparison with existing systems

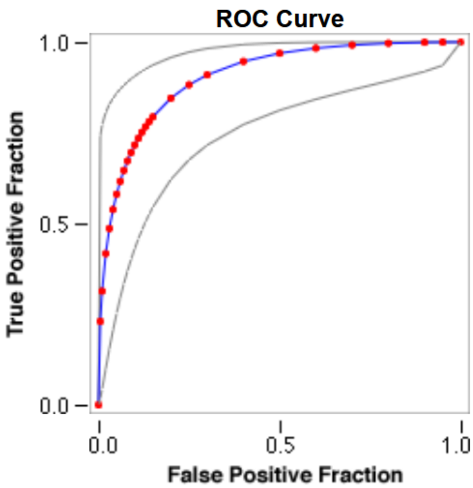

Fig. 5. ROC Curve for the proposed approach

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classification model at different classification thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) as the threshold for classifying positive and negative instances is varied. When interpreting the ROC curve, it's important to note that it is not affected by the imbalance in class distribution. The local and global optima for the ROC curve are defined as follows:

1. Local Optimum: A local optimum refers to a point on the ROC curve where the model achieves the highest TPR for a given FPR or the highest FPR for a given TPR at a specific threshold. It represents a specific trade-off between true positives and false positives based on the chosen threshold.
2. Global Optimum: The global optimum for the ROC curve is the point on the curve that represents the best possible performance of the model across all possible thresholds. In other words, it is the point that maximizes the area under the ROC curve (AUC). A higher AUC indicates better overall discrimination power of the model, with values ranging from 0.5 (no discrimination) to 1.0 (perfect discrimination).

The global optimum of the ROC curve is of particular interest as it provides an aggregate measure of the model's performance across all possible thresholds, capturing its ability to correctly classify positive and negative instances.
However, it's important to note that the selection of a threshold depends on the specific requirements of the problem and the relative importance of false positives and false negatives. The optimal threshold may vary depending on the application and the associated costs or benefits of different types of errors.

The percentage of correct and incorrect predictions made for each group is tabulated. A false positive (FP), a false negative (Fn), a true positive (TP), and a true negative (Tn) are all tracked and accounted for. There are two types of classifications used in this paper's reviews (positive and negative). The performance of the proposed BET-based ABRNN model is measured against that of alternative deep learning strategies. These evaluations are made with the suggested technique for the purpose of assessment.

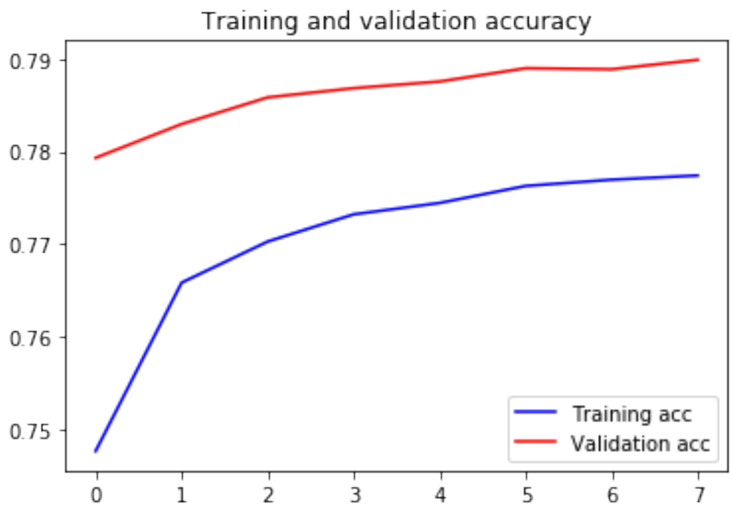

Fig. 6. Training vs Validation accuracy

Common deep learning methods applied to the US-airline review dataset, and their outcomes. Variables including accuracies, recall rates, precisions, f-scores, and areas under the curve (AUCs) are used in the evaluation process. With an accuracy and AUC of 0.95 and 0.97, respectively, the BET-based ABRNN models performed best. Comparatively, the suggested CBRNN model has the greatest accuracy rate (98%) among the other models, while the recall rate (98%) for the Word2vec-based LSTM is identical to that of the BET-based ABRNN model. Training and validation accuracy are compared as shown in Figure 6.

To conclude, the suggested model performed better than the previous models. When compared to previous models, the suggested ABRNN model has the highest recall (91%), f-score (94%), accuracy (90%), and area under the curve (958%). The 96% accuracy rate produced for the BET-based ABRNN is comparable to the LSTM's Word2vec-based precision rate, as seen in the US-airline dataset. The suggested model outperforms all competing baselines in terms of accuracy. The suggested model performed best in terms of recall, f-score, accuracy, and area under the curve. Both these approaches have attained a 98% score on the election reviews dataset, although it has been found that they are completely relying on prediction results. Figure 7 shows the comparison of training and validation loss of the proposed system.

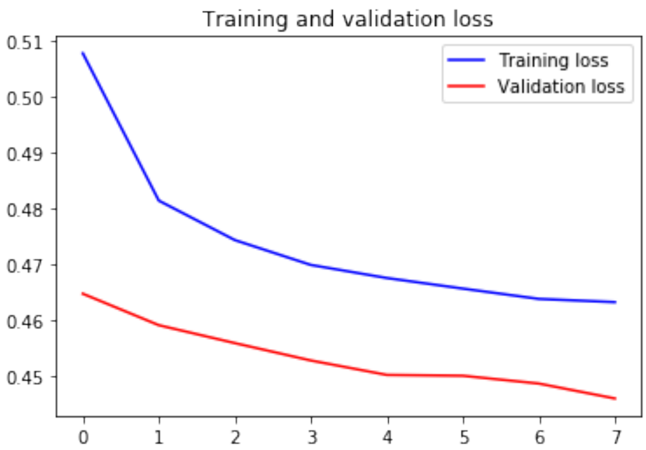

Fig. 7. Training vs validation loss

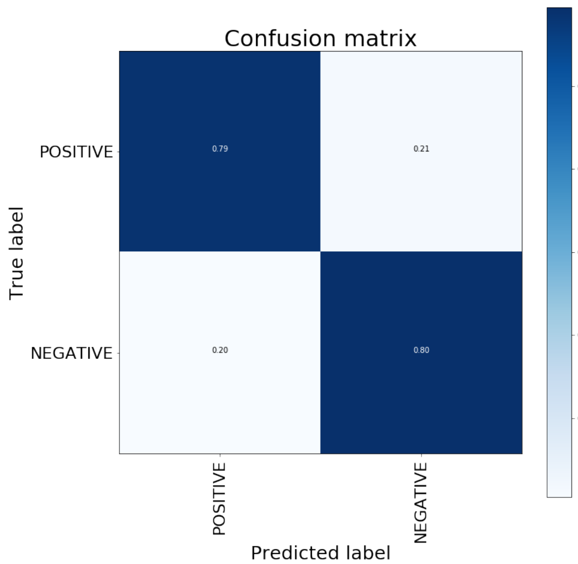

Fig. 8. Confusion matrix for user reviews

Both the accuracy and recall rates of 98% achieved by the BET-based ABRNN model demonstrate its lack of class bias. In addition to having the greatest scores for all matrices, the suggested model also used a BET-based ABRNN model, demonstrating its efficacy. Classifier performance may be seen using the ROC, which is calculated by subtracting the FPR from the TR (TPR). The ROC curve graphic is separated into an x-axis that displays the true positive rate (TPR) and a y-axis that displays the false positive rate (FPR). It has a range of 0–1 and is the best way to determine the SA model that works best. Confusion matrix for user reviews is shown in Figure 8. Kernel distribution of Number of words has been analysed in Figure 9.

Classifier performance is seen to be higher when the skewness of the curve is notably positive. ROC curves for three different models—a Word2vec-based approaches were compared and contrasted. The suggested method's blue curve is more in line with FPR, showing a higher TPR and lower FPR. Confusion matrices show that the suggested model successfully reduced error between anticipated and observed labels. It is a representation of the TP, TN, FP, and FN values that were determined. As a consequence, the experimental findings prove the BET-based ABRNN model's efficacy.

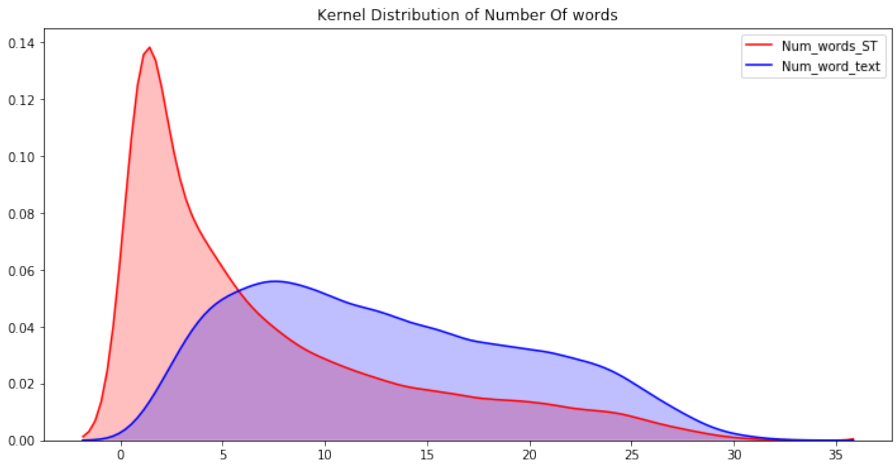

Fig. 9. Kernel distribution of Number of words

The threshold value for the training set in a binary classification task refers to the specific value used to separate predicted positive and predicted negative instances. This threshold determines the point at which the model assigns a sample to either the positive or negative class based on its predicted probability or score. The threshold value is typically chosen based on the specific requirements of the problem and the relative importance of false positives and false negatives. The optimal threshold may vary depending on the application and the associated costs or benefits of different types of errors. In our experiments, a threshold range of 0.5 to 0.55 is used. However, this threshold is not fixed and can be adjusted based on the specific needs of the problem.

5. Conclusion

In this research, we improve upon the state-of-the-art in feature extraction and classification by combining the BET model with the dilated convolutional HLSTM model. For sentence-level labelling, a new BET-based ABRNN SA model has been developed. We used zero-shot BET to annotate our data, and then we used a pre-trained BET model to extract sentence-level semantics and contextual characteristics. The neural network, built using dilated convolution and HLSTM, was then fed the generated contextual embeddings. The purpose of a expanded CNN is to glean information from both a regional and international scale. Using an HLSTM, we may record the sentence's long-term structure. Tests of the proposed hybrid ABRNN model were conducted on four different domain datasets, including evaluations of commercial airlines in the United States, autonomous vehicles, the 2016 presidential election, and visual media. Efficiency, accuracy, f1-score, recall, and area under curve (AUC) are the five statistical metrics used to assess the ABRNN model's efficacy. This data is then compared to the results of other popular embedding methods like glove and word2vec. Improvements of 0.2% in f1-score, 0.3% in accuracy, and 0.4% in AUC were achieved by using the suggested model. The results of the experiments show that the suggested CBRNN model outperforms the alternatives.

References

[1] Kaur, Gagandeep, and Amit Sharma. "A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis." *Journal of Big Data* 10, no. 1 (2023): 5. https://doi.org/10.1186/s40537-022-00680-6

[2] Omran, Thuraya M., Baraa T. Sharef, Crina Grosan, and Yongmin Li. "Transfer learning and sentiment analysis of Bahraini dialects sequential text data using multilingual deep learning approach." *Data & Knowledge Engineering* 143 (2023): 102106. https://doi.org/10.1016/j.datak.2022.102106

[3] Khan, Amjad. "Improved multi-lingual sentiment analysis and recognition using deep learning." *Journal of Information Science* (2023): 01655515221137270. https://doi.org/10.1177/01655515221137270
[4] Kuppusamy, Mouthami, and Anandamurugan Selvaraj. "A novel hybrid deep learning model for aspect based sentiment analysis." *Concurrency and Computation: Practice and Experience* 35, no. 4 (2023): e7538. https://doi.org/10.1002/cpe.7538
[5] Swapnarekha, H., Pandit Byomakesha Dash, and Danilo Pelusi. "An optimistic firefly algorithm-based deep learning approach for sentiment analysis of COVID-19 tweets." *Mathematical Biosciences and Engineering* 20, no. 2 (2023): 2382-2407. https://doi.org/10.3934/mbe.2023112
[6] Shukla, Abhilash, Dhatri Raval, Jaimin Undavia, Nilay Vaidya, Krishna Kant, Sohil Pandya, and Atul Patel. "Deep Learning Applications in Sentiment Analysis." In *ICDSMLA 2021: Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications*, pp. 513-519. Singapore: Springer Nature Singapore, 2023. https://doi.org/10.1007/978-981-19-5936-3_48
[7] Thandaga Jwalanaiah, Swasthika Jain, Israel Jeena Jacob, and Ajay Kumar Mandava. "Effective deep learning based multimodal sentiment analysis from unstructured big data." *Expert Systems* 40, no. 1 (2023): e13096. https://doi.org/10.1111/exsy.13096
[8] Suhartono, Derwin, Kartika Purwandari, Nicholaus Hendrik Jeremy, Samuel Philip, Panji Arisaputra, and Ivan Halim Parmonangan. "Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews." *Procedia Computer Science* 216 (2023): 664-671. https://doi.org/10.1016/j.procs.2022.12.182
[9] Sadigov, Rahim, Elif Yıldırım, Büşra Kocaçınar, Fatma Patlar Akbulut, and Cagatay Catal. "Deep learning-based user experience evaluation in distance learning." *Cluster Computing* (2023): 1-13. https://doi.org/10.1007/s10586-022-03918-3
[10] Hung, Lai Po, and Suraya Alias. "Beyond sentiment analysis: A review of recent trends in text based sentiment analysis and emotion detection." *Journal of Advanced Computational Intelligence and Intelligent Informatics* 27, no. 1 (2023): 84-95. https://doi.org/10.20965/jaciii.2023.p0084
[11] Akter, Sumya, Rumman Ahmed Prodhan, Muhammad Bin Mujib, Md Akhtaruzzaman Adnan, and Tanmoy Sarkar Pias. "Evaluating the Effectiveness of Classification Algorithms for EEG Sentiment Analysis." In *Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022*, pp. 195-212. Singapore: Springer Nature Singapore, 2023. https://doi.org/10.1007/978-981-19-5443-6_17
[12] Dutta, Pijush, Shobhandeb Paul, Korhan Cengiz, Rishabh Anand, and Asok Kumar. "A predictive method for emotional sentiment analysis by deep learning from EEG of brainwave dataset." In *Artificial Intelligence for Neurological Disorders*, pp. 25-48. Academic Press, 2023. https://doi.org/10.1016/B978-0-323-90277-9.00002-X
[13] Adak, Anirban, Biswajeet Pradhan, and Nagesh Shukla. "Sentiment analysis of customer reviews of food delivery services using deep learning and explainable artificial intelligence: Systematic review." *Foods* 11, no. 10 (2022): 1500. https://doi.org/10.3390/foods11101500
[14] Rodrigues, Anisha P., Roshan Fernandes, Adarsh Shetty, Kuruva Lakshmanna, and R. Mahammad Shafi. "Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques." *Computational Intelligence and Neuroscience* 2022 (2022). https://doi.org/10.1155/2022/5211949
[15] Singh, Chetanpal, Tasadduq Imam, Santoso Wibowo, and Srimannarayana Grandhi. "A deep learning approach for sentiment analysis of COVID-19 reviews." *Applied Sciences* 12, no. 8 (2022): 3709. https://doi.org/10.3390/app12083709
[16] Aljedaani, Wajdi, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry." *Knowledge-Based Systems* 255 (2022): 109780. https://doi.org/10.1016/j.knosys.2022.109780
[17] Almalis, Ioannis, Eleftherios Kouloumpris, and Ioannis Vlahavas. "Sector-level sentiment analysis with deep learning." *Knowledge-Based Systems* 258 (2022): 109954. https://doi.org/10.1016/j.knosys.2022.109954
[18] Iqbal, Amjad, Rashid Amin, Javed Iqbal, Roobaea Alroobaea, Ahmed Binmahfoudh, and Mudassar Hussain. "Sentiment analysis of consumer reviews using deep learning." *Sustainability* 14, no. 17 (2022): 10844. https://doi.org/10.3390/su141710844
[19] Saleh, Hager, Sherif Mostafa, Abdullah Alharbi, Shaker El-Sappagh, and Tamim Alkhalifah. "Heterogeneous ensemble deep learning model for enhanced Arabic sentiment analysis." *Sensors* 22, no. 10 (2022): 3707. https://doi.org/10.3390/s22103707
[20] Trisna, Komang Wahyu, and Huang Jin Jie. "Deep learning approach for aspect-based sentiment classification: a comparative review." *Applied Artificial Intelligence* 36, no. 1 (2022): 2014186. https://doi.org/10.1080/08839514.2021.2014186

[21] Al-mashhadani, Mohammed Ibrahim, Kilan M. Hussein, and Enas Tariq Khudir. "Sentiment analysis using optimized feature sets in different facebook/twitter dataset domains using big data." *Iraqi Journal For Computer Science and Mathematics* 3, no. 1 (2022): 64-70. https://doi.org/10.52866/ijcsm.2022.01.01.007

[22] Abbood, Zainab Ali, Baraa Taha Yasen, Mohammed Rashid Ahmed, and Adil Deniz Duru. "Speaker identification model based on deep neural networks." *Iraqi Journal For Computer Science and Mathematics* 3, no. 1 (2022): 108-114. https://doi.org/10.52866/ijcsm.2022.01.01.012

[23] Dineshkumar, R., M. Kalimuthu, K. Deepika, and S. Gopalakrishnan. "Engineering Education with Tool Based Technical Activity (TBTA)." *Journal of Engineering Education Transformations* 36, no. 2 (2022). https://doi.org/10.16920/jeet/2022/v36i2/22166