Multimedia Tools and Applications (2024) 83:18943–18968
https://doi.org/10.1007/s11042-023-16062-w

Emotion prediction for textual data using GloVe based HeBi-CuDNNLSTM model

Dashrath Mahto[1] · Subhash Chandra Yadav[1]

Received: 11 June 2022 / Revised: 15 June 2023 / Accepted: 18 June 2023 /
Published online: 25 July 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

In the digital era, social media websites have emerged as powerful tools for sharing emotions worldwide, and significantly impacting people's social and personal lives. People share their feelings and perspectives in the form of texts, images, emojis, audio, emoticons, and videos. Among the popular social media platforms, Twitter stands out, generating an enormous quantity of unstructured textual data. Due to its strict character limit of 280, Tweets are particularly prone to information diversity, text sarcasm, word sense ambiguity, indirect negation, and other problems. Consequently, identifying and categorizing actual emotions from these textual data becomes a challenging task. To address these challenges; this research paper improved the performance accuracy of the existing HeBiLSTM model by adding one additional Bi-LSTM hidden layer, named as Improved GloVe-HeBiLSTM model. Additionally, the Authors proposed a Hierarchical Bi-CuDNNLSTM with NVIDIA CUDA deep neural network library (named HeBi-CuDNNLSTM) as a practical framework for analyzing actual emotions of textual data. The Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have been trained and validated with *emotions-dataset-for-NLP* dataset and evaluated with various performance measures such as accuracy, recall, precision, and F_1-score. Performance accuracy of the Improved GloVe-HeBiLSTM and the proposed HeBi-CuDNNLSTM frameworks have been compared with existing HeBiLSTM (89%) and recent baseline models, mainly GloVe-CNN-LSTM (92.75%), GloVe-BiLSTM (93.10%), GloVe-CNN-BiLSTM (92.65%), and GloVe-CNN-BiGRU (93.25%). Therefore the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models outperform the existing HeBiLSTM and other baseline models with an accuracy rate of 93.70% and 94.20%, respectively.

Keywords Natural language processing · Deep neural network · Emotion prediction · Sentiment analysis · Bidirectional-LSTM

✉ Dashrath Mahto
dashrathmahto12@gmail.com

Subhash Chandra Yadav
dr.scyadav@cuj.ac.in

[1] Department of Computer Science and Engineering, Central University of Jharkhand, Ranchi, 835205, Jharkhand, India

1 Introduction

Emotion prediction is a method for distinguishing different sorts of human feelings like anger, joy, fear, sadness, and depression. The terms “emotion recognition”, “emotion analysis”, “emotion detection”, and “emotion classification” are frequently used by researchers for emotion prediction [25]. People have grown accustomed to expressing their feelings as access to social media has increased. Social networking sites allow individuals to intensively and freely express their views, thoughts, and beliefs on various subjects. Even for e-commerce businesses, customers provide star ratings and suggestions on their websites for consumer purchases. Therefore service providers and vendors are encouraged to improve their existing systems, products, or services based on customer reviews. Nowadays, almost every sector and organization is going through a digital transformation process that generates a massive amount of heterogeneous data.

With the increased amount of heterogeneous text data, there are many challenges to retrieving significant meaning from these data. Researchers have recently focused on emotion and sentiment analysis of textual data available on microblogging sites because emotion analysis can be helpful for decision-making, product analysis, political promotions, market research, and social media monitoring. Twitter is one of the most famous microblogging sites worldwide, generating enormous text data. Twitter Text data seems crucial to obtain valuable knowledge from extensive data, organizing it, and anticipate user behavior or thoughts.

The majority of existing emotion analysis methods use emotion lexicons and are limited to external resources or manual text pre-processing for more complex features. However, several challenges pertained to the emotion analysis of Twitter text data:

- In emotion analysis tasks, the same word may express opposite orientation when used in different contexts.
- Due to the limited word length of a maximum of 280 characters, information diversity, and sparsed and fragmented nature of Tweets, It is challenging to recognize and categorize actual emotions.
- Several words or even a single sentence can make up a Tweet; it is challenging to discover the implicit contextual features.
- Since abbreviations and word sense ambiguity is used to express emotions instead of formal emotional words. Therefore emotion analysis is challenging.

Considering word sense ambiguities; the Twitter post may contain several terms, as shown in examples 1, 2, and 3.
Example 1: “Flying high”
Example 2: “He was dazzled by a shining light”
Example 3: “Anjali hardly know about research”
These examples demonstrate the importance of word sense ambiguity in expressing emotions in different situations. Since individual word sense ambiguity cannot be manually collected and its features cannot be designed; therefore emotion analysis is challenging. These anomalies related to the existing approaches for emotion prediction from textual data motivated the Authors to propose an enhanced deep neural network-based emotion classification model that can extract different emotional features from text data and accurately predict people’s emotions.

The following are significant contributions made by Authors to resolve the above-mentioned existing challenges:

- Firstly, the Authors improved the performance accuracy of the existing HeBiLSTM [30] model by adding one additional Bi-LSTM hidden layer, named as Improved GloVe-

HeBiLSTM model. Additionally, the Authors proposed a Hierarchical Bi-CuDNNLSTM with NVIDIA CUDA deep neural network library (named HeBi-CuDNNLSTM) as a practical framework for analyzing various emotions of textual data. The Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have utilized the three hierarchical hidden layers to extract a rich number of features from textual data. Both models can efficiently retrieve semantic meaning and sequence characteristics for emotion classification.

- Secondly, the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have been trained, tested, and validated with *emotions-dataset-for-NLP* dataset for analyzing six different emotions such as love, joy, fear, anger, sadness, and surprise.
- Thirdly, The Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have employed a pre-trained GloVe [36] word vectorization technique to portray the preprocessed Twitter text data in the form of vectors of integer data. The GloVe https://github.com/stanfordnlp/glove (accessed on April 25, 2022) creates a word vector space trained on global word-word co-occurrence counts. This approach is pre-trained unlabeled word matrices that could preserve word meaning and learn with a massive set of words.
- Fourthly, the performance of the Improved GloVe-HeBiLSTM and the proposed HeBi-CuDNNLSTM models have been evaluated using different performance measures like accuracy, recall, precision, and F_1 Score.
- Lastly, the experimental outcomes of several baseline deep learning models and existing HeBiLSTM [30] benchmarks confirm the effectiveness of the proposed HeBi-CuDNNLSTM and Improved GloVe-HeBiLSTM models.

The research article is organized as follows: Section 2 discussed a literature overview of related studies, Background about word embedding, and deep neural networks. Then, Section 3 has described the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM frameworks. Next, Section 4 has described the experimental setup, dataset, parameter settings, baseline models, and performance evaluation metrics. Then, Section 5 analyzed the experimental results and compared the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models with existing HeBiLSTM and other recent baseline models. Finally, Section 6 summarizes the work carried out during the research study and also provides future scope in the field of emotion prediction.

2 Background

This section contains a critical analysis of related research work in the field of deep neural networks (DNN) and other existing state-of-the-art models for emotion prediction [26]. The section also comprises the basic concept of word embedding and DNN used during the research work.

2.1 Related research

Jayakrishnan et al. [22] used a supervised ML-based SVM model for emotion classification from the Malayalam dataset that contains 500 sentences. The SVM used negation level, POS tagging, and N-gram syntactic features for better accuracy improvement. The accuracy of SVM for happy, sad, angry, fear, and surprise are 94%, 92%, 90%, 93%, and 90% respectively. Ragheb et al. [39] used an attention-based BiLSTM approach for emotion classification from

SemEval2019 Task 3 dataset [12]. The model used a transfer learning approach for word embedding. The micro-F_1 score of the proposed model is 75.82%. Ma et al. suggested a Bi-LSTM approach [29] with attention mechanism and GloVe embedding for capturing syntactic and semantic knowledge and for predicting emotion from SemEval2019 Task 3 dataset. The suggested BiLSTM model is restricted to only classifying three happy, sad, and angry emotions of the SemEval2019 dataset. The accuracy of BiLSTM for happy, sad, and angry emotions is 71.38%, 80.88%, and 75% respectively. Ahmad et al. [2] suggested a hybrid CNN and BiLSTM framework to classify sentiments and emotions from three different English (SemEval-2018, Emotion in Text) and Hindi (Hindi reviews) datasets. The model used cross-lingual embeddings, and transfer learning (TL) approaches to enhance the accuracy of the classification. Batbaatar et al. [5] suggested the SENN model that uses the CNN network for emotion encoding and the BiLSTM network for semantic encoding. The proposed SENN model implemented with emotion-annotated ten datasets of different domains to classify different emotions. The SENN model is trained with three different word vectorization techniques: GloVe, FastText, and Word2Vec. Abid et al. [1] devised a method for concatenating decentralized word vectors (DWRs) using a weighted framework based on RNN variations in conjunction with CNN, which have used for weighted attentive pooling (WAP). Semantic & syntactic rules and out-of-vocabulary (OOV) words have been handled. The experimental analysis has shown an average accuracy of 89.67%. However, it had a flaw in insufficient extracting features, which resulted in analytical errors. Sundaram et al. [44] used the TF-IDF algorithm for classifying six emotional classes of *emotions-dataset-for-NLP* dataset. The overall accuracy of the TF-IDF algorithm with *emotions-dataset-for-NLP* dataset is 85%. Mahto and Yadav [30] suggested an enhanced DNN-based Hierarchical Bi-LSTM model for emotion classification from text data of microblogging sites like Twitter. The Hierarchical Bi-LSTM model has been trained and tested with *emotions-dataset-for-NLP* dataset for classifying six emotional classes: love, anger, sadness, surprise, joy, and fear. The Authors have compared the suggested model with the existing hybrid CNN-LSTM model, which outperformed with an accuracy of 89%.

As reviewed above the literary works carried out by various researchers. It has been observed that there are various machine learning and deep learning approaches have been used to predict the emotions of textual data. However, the study also reveals that there is a quite need for improvement and development of an effective and efficient classification model for analyzing different emotions of textual data. Jayakrishnan et al. [22], Ma et al. [29], and Ragheb et al. [39] used a limited number of emotions for analysis and the performance accuracy is not remarkable [30, 44]. The existing LSTM, Bi-LSTM, CNN-LSTM, GRU, and Bi-GRU, suffer from a vanishing gradient problem during deep back-propagation. This can happen when the gradients become very small during back-propagation, making it difficult for the model to learn long-term textual data dependencies. These models are less capable to handle class imbalance problems of datasets and struggle to understand the contextual information of the text data. Therefore, an enhanced deep neural network is required to solve these problems.

2.2 Word embedding

Many research studies have described the utilization of word representation techniques to capture syntax and semantics of text data. The Word Net [4] is one of the information repositories used in the word representation approaches that is a linguistic dataset tagged by linguists. This method, however, requires moment manual annotation of word links, and the outcomes are susceptible to input subjectivity, restricting sight word link prediction. The bag-of-words method, or corpus-based one-hot description, is a traditional word descriptor in which each sentence is represented by a scalar [46]. Another representation method is the coordination method matrix, which evaluates the semantic relationship of a set of terms in a sentence with a specific window size. Word embedding [6, 50] is a distributed word learning method that uses neural network (NN) topology and has also been developed in recent years to express words as real-valued low-dimensional dense vectors. Word embedding is a powerful method for extracting semantic and syntactic meaning from massive data sets. The NN language model [6] is a pioneering project that teaches word vectors from word contexts. Word2vec [32, 33] is a commonly used word insertion learning method that attempts to use a single and simple neural network architecture to predict the target word according to the situation. GloVe [36] is another sentence insertion learning method that creates word insertions by using general statistics to detect possible co-occurrence patterns in the corpus [7, 8, 10, 11]. In addition to the available sentiment analysis [28] mentioned above, some researchers also recommend updating previously trained word vectors by providing learning resources to improve specific subsequent applications [16, 50]. The attention mechanism has recently been added to word embedding systems for visual words, such as BERT [15, 49], GPT-2 [38]. BERT and GPT-2 are converter models [47], with BERT relying entirely on the converter's encoder and GPT-2 relying purely on the converter's decoding. The identity layer has been used by the encoder and decoder in the conversion to determine the interest weights for any statuses that aren't displayed. The classification algorithm is a learner-centered method in CNN architectures for different NLP applications.

2.3 Deep Neural Network (DNN)

Deep learning systems offer economical learning ways for various uses and analyses. Deep NNs [9] are a gaggle of feed upstream NNs that use the approximating a theorem to estimate given enough cells, any sophisticated having full time may be performed to create a NN model [14, 51]. Neural networks won't solve a variety of information science challenges [19]. RNN is effectively used with textual information. A basic RNN is a selective memory that optimizes models via reverse computing using back-propagation across time but cannot handle big datasets. The LSTM is an extension model of RNN that improves the shortcomings of RNN. The LSTM utilizes three gates to extract local meanings over time an input gate, a forget gate, and an output gate. The input nodes used here will recall the information if the present word is not the same as the initial letters, and the input gate will choose the current frame's data created. These three gateways permit data to join, exit, or erase, avoiding vanishing gradients, which can be problematic in long sequential texts. To form a document sentiment classifier, Rao et al. [42] suggested employing an LSTM network with phrase representations. Their projected model outperformed different progressive models on three publicly available information analysis datasets. To quantify sentiment strength for specific text data, Wang et al. [48] suggested the stacked residual LSTM framework. As per their experimental outcome, LSTM with additional stack layers can do adequate classification accuracy. Bi-directional LSTM has been utilized in numerous research to retrieve useful characteristics. The BiLSTM contains one forward LSTM layer and one backward LSTM layer for feature extraction for past and future timestamps.

3 Proposed GloVe-based HeBi-CuDNNLSTM framework for emotion prediction

This research article improved the performance of the existing HeBiLSTM model by adding one additional Bi-LSTM hidden layer, named Improved GloVe-HeBiLSTM model and additionally, the Authors proposed an enhanced deep learning GloVe-based HeBi-CuDNNLSTM framework for classifying six emotions such as love, anger, sadness, surprise, joy, and fear. The Improved GloVe-based HeBiLSTM and proposed HeBi-CuDNNLSTM models have the same hierarchical hidden layers for features extraction and selection with the same parameter settings as shown in Table 1. The only difference is that Improved GloVe-HeBiLSTM used LSTM instead of CuDNNLSTM and can run on CPU (Central Processing Unit), TPU (Tensor Processing Units), and GPU (Graphics Processing Unit). The proposed HeBi-CuDNNLSTM framework used CuDNNLSTM and can only run on GPU. Therefore the performance of the proposed framework is better than Improved GloVe-HeBiLSTM. The Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM frameworks have similar workflow as described in Eqs. (3)–(41) [30, 54]. The Authors described the workflow of the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM frameworks for emotion predictions in four stages. In the first stage, the Authors performed pre-processing of raw Twitter texts of *emotions-dataset-for-NLP* dataset. In the second stage, the Authors applied the pre-trained GloVe (Glove.840B.300D) word vectorization technique in the pre-processed dataset to generate word vector representations for word-word co-occurrence. The numerical value of each input word or token is embedded with a 300-dimensional GloVe vector. This GloVe embedding technique is considered more efficient and powerful than other word2vec embedding techniques. In the third stage, the Authors built three-layered architecture of Hierarchical Bi-CuDNNLSTM as hidden layers to retrieve contextual features necessary for describing text accurately. To extract comprehensive context data from past and future situations, HeBi-CuDNNLSTM used three interconnected CuDNNLSTMs in the forward direction and three CuDNNLSTMs in the backward direction; since the CuDNNLSTM is more powerful and faster than LSTM. The CuDNNLSTM only runs on GPU and always uses TensorFlow as the backend. In the fourth stage, the output of the last Bi-CuDNNLSTM hidden layer has received by a fully-connected first dense layer. This fully-connected first dense layer used the swish [40] activation function to transform the hierarchical bidirectional network into high-level emotion representations and passes the outcome to the last dense layer. This last dense layer uses a softmax activation function for classifying six different emotional classes. The Layered structure of the Improved Improved GloVe-HeBiLSTM and proposed Hierarchical Bidirectional CuDNNLSTM Frameworks have been shown in Fig. 1. This layered structure improves the effectiveness of the proposed system for emotion recognition, allowing for the precise classification of emotional responses. The following subsections comprise a detailed discussion of the four stages of Improved GloVe-HeBiLSTM and Proposed HeBi-CuDNNLSTM frameworks.

3.1 Data pre-processing

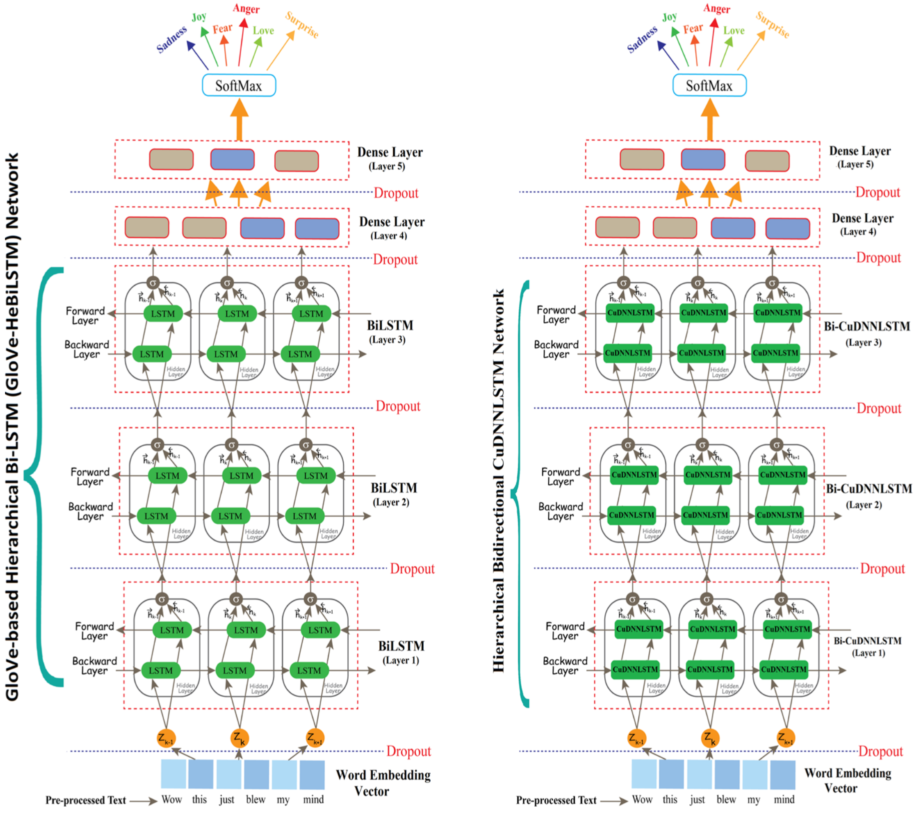
The Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have been trained and tested on “emotions-dataset-for-NLP” dataset that is available on Kaggle. The primary purpose of the pre-processing stage is to eliminate undesired distortion, inconsistencies, noises, anomalies, unwanted symbols, hashtags, URLs, and duplication from the raw text data. The data preparation is required to clean the data by making it appropriate for

Table 1 Parameter settings for experimental GloVe-based deep neural network models

Parameters	GloVe-CNN-LSTM	GloVe-CNN-BiLSTM	GloVe-CNN-BiGRU	GloVe-BiLSTM	Improved GloVe-HeBiLSTM	Proposed HeBi-CuDNNLSTM Framework
Input Dimensions (Vocabulary Size)	15065					
GloVe Embedding Dimensions	300					
Maximum Input Length	35					
Convolutional Layer	filters=32, kernel_size=3, activation=swish			NA		
Max Pooling Layer	pool_size=3			NA		
Output shape (hidden layer 1)	Conv1D (None, 33, 32)			Bidirectional-LSTM (None, 512)	Bidirectional-LSTM (None, 35, 512)	Bidirectional-CuDNNLSTM (None, 35, 512)
Output shape (hidden layer 2)	MaxPooling1D (None, 11, 32)			NA	Bidirectional-LSTM (None, 35, 512)	Bidirectional-CuDNNLSTM (None, 35, 512)
Output shape (hidden layer 3)	LSTM (None, 256)	Bidirectional-LSTM (None, 512)	Bidirectional-GRU (None, 512)	NA	Bidirectional-LSTM (None, 512)	Bidirectional-CuDNNLSTM (None, 512)

Table 1 continued

Parameters	GloVe-CNN-LSTM	GloVe-CNN-BiLSTM	GloVe-CNN-BiGRU	GloVe-BiLSTM	Improved GloVe-HeBiLSTM	Proposed HeBi-CuDNNLSTM Framework
Output shape (dense layer 1)	(None, 256)					
Output shape (dense layer 2)	(None, 6)					
Batch Size	128					
No. of Epochs	15					
Dropouts	0.3					
Activation Function (Dense layer 1)	Swish					
Activation Function (Dense layer 2)	Softmax					
Optimizer	Adam					
Learning rate	0.001					
Loss function	categorical_crossentropy					



(a) Improved GloVe-HeBiLSTM

(b) Proposed HeBi-CuDNNLSTM

Fig. 1 Layered structure of **a** Improved GloVe-HeBiLSTM and **b** Proposed HeBi-CuDNNLSTM model

further analysis. The Authors have performed the following pre-processing tasks to clean the raw text data including removal of punctuations, elimination of stopwords, stemming using PorterStemmer function, Lemmatization using WordNet, removal of numerical values and conversion of text corpus into lowercase. The sequence of pre-processing techniques that have been applied “emotions-dataset-for-NLP” dataset is shown in Fig. 2.

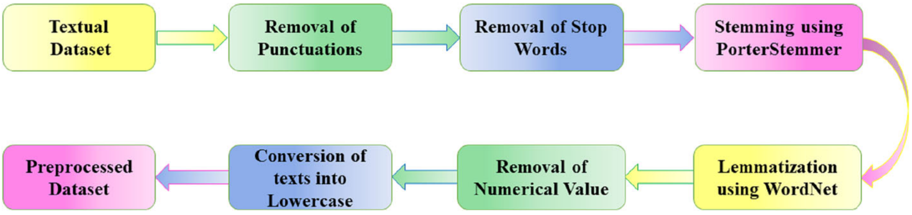

Textual Dataset
Removal of Punctuations
Removal of Stop Words
Stemming using PorterStemmer
Lemmatization using WordNet
Removal of Numerical Value
Conversion of texts into Lowercase
Preprocessed Dataset

Fig. 2 Sequence of preprocessing techniques used for emotions-dataset-for-NLP dataset

3.2 Word vectorization

In this stage, the pre-processed data is taken as input and then segmented into individual words or tokens. The pre-trained GloVe vectorization algorithm generates a vector matrix of numerical values for every token. If every text data of p-words is expressed as $D = \{w_1, w_2, \ldots, w_p\}$, then **m**-dimensional vector representation is generated for every word, and the input text is specified as:

$$D = \{w_1, w_2, ..., w_p\} \in V^{p*m} \tag{1}$$

Since the length of every input text data has a variable size length, the text size length must be equal to size **s**. If the size of the text is smaller than **s**, then the size of the text gets increased by applying a zero-padding technique, and if the size of the text is larger than the specified size **s**, then it will be shortened; as a result, every text content has the same vector size. In this article, the fixed maximum input length size of text is 35 words or tokens. Representation of each text content of **s** dimension is as given in Eq. (2) [31, 45]:

$$D = \{w_1, w_2, ..., w_p\} \in V^{s*m} \tag{2}$$

Here V is the vector matrix of numerical values for every word in the input text data.

3.3 Hierarchical Bi-CuDNNLSTM Hidden Layers for features extraction and selection

In this stage, the Hierarchical Bi-CuDNNLSTM Hidden Layers used word embedding vectors of numeric values as input. The Hierarchical Bi-CuDNNLSTM framework used three Bidirectional CuDNNLSTM layers to classify different emotions from textual data. These hierarchical Bi-CuDNNLSTM layers can retrieve rich numbers of contextual data from future and past time frames compared to Bi-LSTM or Bi-CuDNNLSTM. On the other hand, HeBi-CuDNNLSTM has more top layers to perform additional feature extraction processes than Bi-LSTM or Bi-CuDNNLSTM, which only has a single hidden layer for every dimension to extract essential features. Figure 3 depicts the three-layered architecture of the Hierarchical Bi-CuDNNLSTM Hidden Layers.

For time sequence K, the input sequence $\{z_1, z_2, ..., z_K\}$ passes through hidden units in forward direction $\{\overrightarrow{a_1}, \overrightarrow{a_2}, ..., \overrightarrow{a_K}\}$ to receive every previous iteration phases and hidden units in the reverse direction $\{\overleftarrow{a_1}, \overleftarrow{a_2}, ..., \overleftarrow{a_K}\}$ to receive all upcoming iterations. Following that, the top hidden units have used the results of the bottom hidden units at every time interval as sources to retrieve additional features. The top levels of forward hidden units are denoted by $\{\overrightarrow{b_1}, \overrightarrow{b_2}, ..., \overrightarrow{b_K}\}$ and $\{\overrightarrow{c_1}, \overrightarrow{c_2}, ..., \overrightarrow{c_K}\}$, whereas the upper layers of reverse hidden units are denoted by $\{\overleftarrow{b_1}, \overleftarrow{b_2}, ..., \overleftarrow{b_K}\}$ and $\{\overleftarrow{c_1}, \overleftarrow{c_2}, ..., \overleftarrow{c_K}\}$. Finally, output layers combine the hidden vectors of the three above stages as its outcome as represented in Eq. (39). Each node of every hidden layer of the proposed framework represents a CuDNNLSTM cell with an input gate as i_k, a forget gate as f_k, and an output gate o_k. Authors select one of the alternatives of CuDNNLSTM to compute hidden states $\overrightarrow{a_k}$, $\overrightarrow{b_k}$, $\overrightarrow{c_k}$, $\overleftarrow{a_k}$, $\overleftarrow{b_k}$, and $\overleftarrow{c_k}$ at every layer for every time step **k**.

The hidden state $\overrightarrow{a_k}$ of the first forward layer is expressed in the Equations below:

Input Gate ($i_k^{\overrightarrow{a}}$):

$$i_k^{\overrightarrow{a}} = \sigma(Y_i^{\overrightarrow{a}} z_k + X_i^{\overrightarrow{a}} \overrightarrow{a}_{k-1} + b_i^{\overrightarrow{a}}) \tag{3}$$

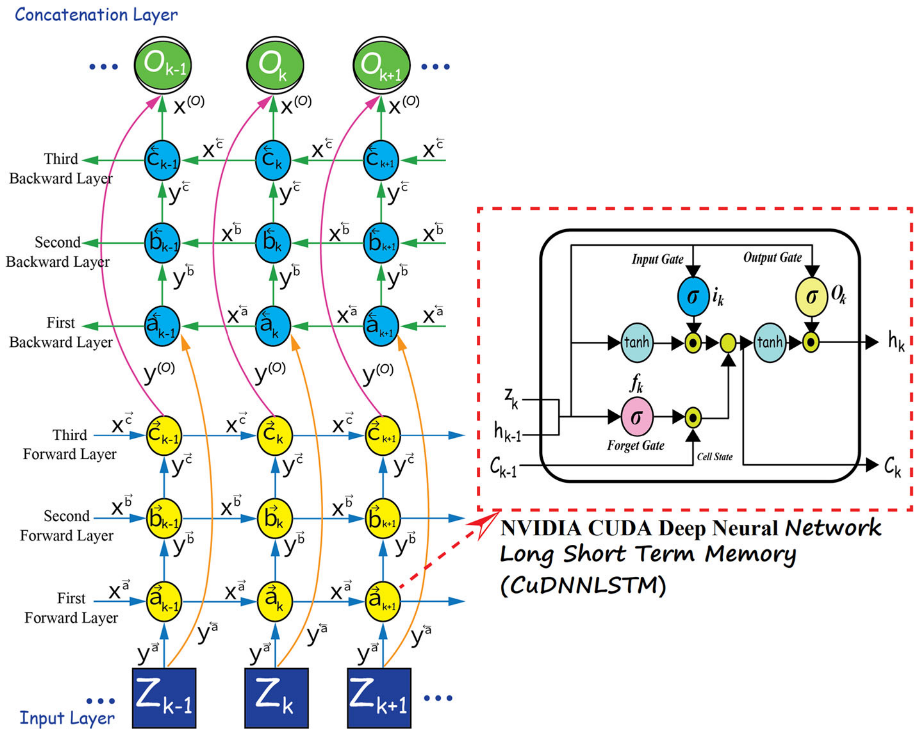


Fig. 3 Three-layered architecture of Hierarchical Bi-CuDNNLSTM Hidden Layers

Forget Gate ($f_k^{\overrightarrow{a}}$):

$$f_k^{\overrightarrow{a}} = \sigma(Y_f^{\overrightarrow{a}} z_k + X_f^{\overrightarrow{a}} \overrightarrow{a}_{k-1} + b_f^{\overrightarrow{a}}) \tag{4}$$

Output Gate ($o_k^{\overrightarrow{a}}$)

$$o_k^{\overrightarrow{a}} = \sigma(Y_o^{\overrightarrow{a}} z_k + X_o^{\overrightarrow{a}} \overrightarrow{a}_{k-1} + b_o^{\overrightarrow{a}}) \tag{5}$$

Cell State ($C_k^{\overrightarrow{a}}$):

$$u_k^{\overrightarrow{a}} = \tanh(Y_u^{\overrightarrow{a}} z_k + X_u^{\overrightarrow{a}} \overrightarrow{a}_{k-1} + b_u^{\overrightarrow{a}}) \tag{6}$$

$$C_k^{\overrightarrow{a}} = i_k^{\overrightarrow{a}} \odot u_k^{\overrightarrow{a}} + f_k^{\overrightarrow{a}} \odot C_{k-1}^{\overrightarrow{a}} \tag{7}$$

Cell Outputs ($\overrightarrow{a_k}$):

$$\overrightarrow{a_k} = o_k^{\overrightarrow{a}} \odot \tanh(C_k^{\overrightarrow{a}}) \tag{8}$$

Similarly, the next hidden state $\overrightarrow{b_k}$ of the second forward layer is expressed in the Equations below:

$$i_k^{\overrightarrow{b}} = \sigma(Y_i^{\overrightarrow{b}} \overrightarrow{a_k} + X_i^{\overrightarrow{b}} \overrightarrow{b}_{k-1} + b_i^{\overrightarrow{b}}) \tag{9}$$

$$f_k^{\overrightarrow{b}} = \sigma(Y_f^{\overrightarrow{b}} \overrightarrow{a_k} + X_f^{\overrightarrow{b}} \overrightarrow{b}_{k-1} + b_f^{\overrightarrow{b}}) \tag{10}$$

$$o_k^{\overrightarrow{b}} = \sigma(Y_o^{\overrightarrow{b}} \overrightarrow{a_k} + X_o^{\overrightarrow{b}} \overrightarrow{b}_{k-1} + b_o^{\overrightarrow{b}}) \tag{11}$$

$$u_k^{\overrightarrow{b}} = \tanh(Y_u^{\overrightarrow{b}} \overrightarrow{a_k} + X_u^{\overrightarrow{b}} \overrightarrow{b}_{k-1} + b_u^{\overrightarrow{b}}) \tag{12}$$

$$C_k^{\overrightarrow{b}} = i_k^{\overrightarrow{b}} \odot u_k^{\overrightarrow{b}} + f_k^{\overrightarrow{b}} \odot C_{k-1}^{\overrightarrow{b}} \tag{13}$$

$$\overrightarrow{b_k} = o_k^{\overrightarrow{b}} \odot \tanh(C_k^{\overrightarrow{b}}) \tag{14}$$

Similarly, the next hidden state $\overrightarrow{c_k}$ of the third forward layer is expressed in the Equations below:

$$i_k^{\overrightarrow{c}} = \sigma(Y_i^{\overrightarrow{c}} \overrightarrow{b_k} + X_i^{\overrightarrow{c}} \overrightarrow{c}_{k-1} + b_i^{\overrightarrow{c}}) \tag{15}$$

$$f_k^{\overrightarrow{c}} = \sigma(Y_f^{\overrightarrow{c}} \overrightarrow{b_k} + X_f^{\overrightarrow{c}} \overrightarrow{c}_{k-1} + b_f^{\overrightarrow{c}}) \tag{16}$$

$$o_k^{\overrightarrow{c}} = \sigma(Y_o^{\overrightarrow{c}} \overrightarrow{b_k} + X_o^{\overrightarrow{c}} \overrightarrow{c}_{k-1} + b_o^{\overrightarrow{c}}) \tag{17}$$

$$u_k^{\overrightarrow{c}} = \tanh(Y_u^{\overrightarrow{c}} \overrightarrow{b_k} + X_u^{\overrightarrow{c}} \overrightarrow{c}_{k-1} + b_u^{\overrightarrow{c}}) \tag{18}$$

$$C_k^{\overrightarrow{c}} = i_k^{\overrightarrow{c}} \odot u_k^{\overrightarrow{c}} + f_k^{\overrightarrow{c}} \odot C_{k-1}^{\overrightarrow{c}} \tag{19}$$

$$\overrightarrow{c_k} = o_k^{\overrightarrow{c}} \odot \tanh(C_k^{\overrightarrow{c}}) \tag{20}$$

Similarly, the next hidden state $\overleftarrow{a_k}$ of the first backward layer is expressed in the Equations below:

$$i_k^{\overleftarrow{a}} = \sigma(Y_i^{\overleftarrow{a}} z_k + X_i^{\overleftarrow{a}} \overleftarrow{a}_{k+1} + b_i^{\overleftarrow{a}}) \tag{21}$$

$$f_k^{\overleftarrow{a}} = \sigma(Y_f^{\overleftarrow{a}} z_k + X_f^{\overleftarrow{a}} \overleftarrow{a}_{k+1} + b_f^{\overleftarrow{a}}) \tag{22}$$

$$o_k^{\overleftarrow{a}} = \sigma(Y_o^{\overleftarrow{a}} z_k + X_o^{\overleftarrow{a}} \overleftarrow{a}_{k+1} + b_o^{\overleftarrow{a}}) \tag{23}$$

$$u_k^{\overleftarrow{a}} = \tanh(Y_u^{\overleftarrow{a}} z_k + X_u^{\overleftarrow{a}} \overleftarrow{a}_{k+1} + b_u^{\overleftarrow{a}}) \tag{24}$$

$$C_k^{\overleftarrow{a}} = i_k^{\overleftarrow{a}} \odot u_k^{\overleftarrow{a}} + f_k^{\overleftarrow{a}} \odot C_{k-1}^{\overleftarrow{a}} \tag{25}$$

$$\overleftarrow{a_k} = o_k^{\overleftarrow{a}} \odot \tanh(C_k^{\overleftarrow{a}}) \tag{26}$$

Similarly, the next hidden state $\overleftarrow{b_k}$ of the second backward layer is expressed in the Equations below:

$$i_k^{\overleftarrow{b}} = \sigma(Y_i^{\overleftarrow{b}} \overleftarrow{a_k} + X_i^{\overleftarrow{b}} \overleftarrow{b}_{k+1} + b_i^{\overleftarrow{b}}) \tag{27}$$

$$f_k^{\overleftarrow{b}} = \sigma(Y_f^{\overleftarrow{b}} \overleftarrow{a_k} + X_f^{\overleftarrow{b}} \overleftarrow{b}_{k+1} + b_f^{\overleftarrow{b}}) \tag{28}$$

$$o_k^{\overleftarrow{b}} = \sigma(Y_o^{\overleftarrow{b}} \overleftarrow{a_k} + X_o^{\overleftarrow{b}} \overleftarrow{b}_{k+1} + b_o^{\overleftarrow{b}}) \tag{29}$$

$$u_k^{\overleftarrow{b}} = \tanh(Y_u^{\overleftarrow{b}} \overleftarrow{a_k} + X_u^{\overleftarrow{b}} \overleftarrow{b}_{k+1} + b_u^{\overleftarrow{b}}) \tag{30}$$

$$C_k^{\overleftarrow{b}} = i_k^{\overleftarrow{b}} \odot u_k^{\overleftarrow{b}} + f_k^{\overleftarrow{b}} \odot C_{k-1}^{\overleftarrow{b}} \tag{31}$$

$$\overleftarrow{b_k} = o_k^{\overleftarrow{b}} \odot \tanh(C_k^{\overleftarrow{b}}) \tag{32}$$

Similarly, the last hidden state $\overleftarrow{c_k}$ of the third backward layer is expressed in the Equations below:

$$i_k^{\overleftarrow{c}} = \sigma(Y_i^{\overleftarrow{c}} \overleftarrow{b_k} + X_i^{\overleftarrow{c}} \overleftarrow{c}_{k+1} + b_i^{\overleftarrow{c}}) \tag{33}$$

$$f_k^{\overleftarrow{c}} = \sigma(Y_f^{\overleftarrow{c}} \overleftarrow{b_k} + X_f^{\overleftarrow{c}} \overleftarrow{c}_{k+1} + b_f^{\overleftarrow{c}}) \tag{34}$$

$$o_k^{\overleftarrow{c}} = \sigma(Y_o^{\overleftarrow{c}} \overleftarrow{b_k} + X_o^{\overleftarrow{c}} \overleftarrow{c}_{k+1} + b_o^{\overleftarrow{c}}) \tag{35}$$

$$u_k^{\overleftarrow{c}} = \tanh(Y_u^{\overleftarrow{c}} \overleftarrow{b_k} + X_u^{\overleftarrow{c}} \overleftarrow{c}_{k+1} + b_u^{\overleftarrow{c}}) \tag{36}$$

$$C_k^{\overleftarrow{c}} = i_k^{\overleftarrow{c}} \odot u_k^{\overleftarrow{c}} + f_k^{\overleftarrow{c}} \odot C_{k-1}^{\overleftarrow{c}} \tag{37}$$

$$\overleftarrow{c_k} = o_k^{\overleftarrow{c}} \odot \tanh(C_k^{\overleftarrow{c}}) \tag{38}$$

The final hidden layer output O_k is represented as the integrated hidden vector form of the third forward layer $\overrightarrow{c_k}$ and the third backward layer $\overleftarrow{c_k}$ at every time step k, represented as [30, 54]:

$$O_k = Y^{(O)} \overrightarrow{c_k} + X^{(O)} \overleftarrow{c_k} + b^{(O)} \tag{39}$$

3.4 Dense layer and emotion prediction

In this stage, the output O_k of the last Bi-CuDNNLSTM hidden layer has been passed to a fully-connected first dense layer. This fully-connected layer uses the swish [40] activation function to transform the hierarchical bidirectional network into high-level emotion representations. The output of the first dense layer is expressed in Eq. (40).

$$O_k^{Dense_1} = swish\{w_i.O_k + b_i\} \tag{40}$$

Where weight (w_i) and bias (b_i) are parameters, and O_k is the feature map generated from the last Bi-CuDNNLSTM Hidden Layer. The obtained features $O_k^{Dense_1}$ have passed to the last dense layer that uses a softmax activation function to classify six different emotions. The predicted outcome of the softmax function is expressed in Equation (41). The output of anger, fear, joy, love, sadness, and surprise emotions are labeled with 0, 1, 2, 3, 4, and 5, respectively.

$$O_k^{Dense_2} = softmax\{w_i.O_k^{Dense_1} + b_i\} \tag{41}$$

Categorical cross-entropy has been used as a loss function, and Adam [24] optimizer with a learning rate of 0.001 is used to optimize the parameters of the suggested framework. The dropout [21] rate of 30% have used before and after each hidden layer to avoid the over-fitting problem.

4 Research experiments

In this section, the Authors discussed the experimental setup, dataset, parameter settings, and baseline models including GloVe-CNN-LSTM, GloVe-CNN-BiLSTM, GloVe-CNN-BiGRU, and GloVe-BiLSTM and also introduced performance evaluation metrics that are followed by the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models.

4.1 Experimental setup

Various methods, tools, and packages are available to develop deep neural network models. Keras [17] is used with TensorFlow as the backend in this research work. Keras is known to be one of the best tools of Python for developing deep neural network models. All the research experiments executed in Google collaboratory and Jupiter notebook on Ubuntu 18.04.6 LTS with Intel® Core™ i5-4440 CPU @ 3.10GHz × 4, AMD® Cedar Graphics, and 4 GB RAM as hardware and software support.

4.2 Data extraction

The dataset (tweet corpus) (https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp) is suitable for emotion recognition using Tweets from Twitter. The emotion-dataset-for-NLP dataset is publicly available on Kaggle. That contains 20000 rows and two columns with six labels. Whereas 16000 rows have been used for training, 2000 rows for testing, and 2000 rows for validation. There are 719 surprises, 1641 love, 2373 fear, 2709 anger, 5797 sadness, and 6761 rows for joy in this dataset.

4.3 Parameter settings

The parameter settings of Improved GloVe-HeBiLSTM, the Proposed framework, and other GloVe-based baseline deep learning models have shown in Table 1. The Improved GloVe-HeBiLSTM and the proposed HeBi-CuDNNLSTM models have used 300-dimensional GloVe vector embedding as input for the first hidden layer. The fixed maximum input length size of text is 35 words, the rate of dropouts is 30%, each hidden layer Bi-CuDNNLSTM output size is 512, and there are 15 epochs. The batch size of each epoch is 128, "swish" activation function used in a fully-connected first dense layer. The softmax activation function is used in the last dense layer to classify different emotional outputs.

4.4 Baseline models

Emotion classification studies have drawn attention among researchers during the past few years. The Authors examined standard Machine Learning approaches and determined that they were inefficient. As part of the experiments, the Authors also examined whether the proposed framework can predict emotions better than existing deep learning models. The Authors compared the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models with recent baseline models to verify the performance accuracy. Some of the recent baseline models have been discussed as mentioned below.

- **GloVe-CNN-LSTM:** Compared with the proposed framework, GloVe-CNN-LSTM [3, 20, 41, 52, 53] have the same GloVe word embedding as the input layer and dense layer with softmax function as the output layer. The difference is that the GloVe-CNN-LSTM has only one forward LSTM layer. And the convolutional layer of GloVe-CNN-LSTM has been used with 32 filters, three kernels, and the swish activation function to find the most important features in the vector received from the embedding layer. By converting every kernel size into a single output of the highest value, the 1-dimensional max-pooling layer decreases the dimensions of its input using pool size 3.
- **GloVe-CNN-BiLSTM:** Compared with the proposed framework, GloVe-CNN-BiLSTM [23, 27, 31, 45] have the same GloVe word embedding as input layer and dense layer with a softmax activation as output layer. The difference is that the convolutional layer of GloVe-CNN-BiLSTM has used 32 filters and three kernels. And swish activation to find the most important features in the vector-matrix received from the embedding layer and the 1-dimensional max-pooling layer of GloVe-CNN-BiLSTM have three pools. The

GloVe-CNN-BiLSTM has one forward LSTM layer and one backward LSTM layer for feature extraction for past and future timestamps.

- **GloVe-CNN-BiGRU:** Compared with proposed framework, GloVe-CNN-BiGRU [18, 43] have same parameter settings as GloVe-CNN-LSTM. But GloVe-CNN-BiGRU has one forward Gated Recurrent Unit (GRU) layer and one backward GRU layer used for features extraction for past and future timestamps.
- **GloVe-BiLSTM:** Compared with the proposed framework, GloVe-BiLSTM [13, 34, 35, 37] also has the same GloVe word embedding as input and dense layer as output with a softmax activation function. The only difference is that GloVe-BiLSTM used LSTM instead of CuDNNLSTM, and it does not contain any hierarchical layers, and GloVe-BiLSTM contains one forward and one backward layer to extract different features from text data.

  Therefore the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models outperform the baseline models.

4.5 Performance evaluation metrics

To correlate the experimental outcomes of the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models with existing HeBiLSTM and other recent baseline models, the accuracy value is used as a primary performance indicator. Additional performance measures, including recall, precision, and F1-measure have been used to measure the performance of the Improved GloVe-HeBiLSTM and proposed models.

$$Pre = \frac{TP}{TP + FP} \tag{42}$$

$$Rec = \frac{TP}{TP + FN} \tag{43}$$

$$Acc = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{44}$$

$$f_1 - Score = \frac{2 * Pre * Rec}{Pre + Rec} \tag{45}$$

Pre, Rec, Acc, TP, TN, FP, and FN are Precision, Recall, Accuracy, True Positive, True Negative, False Positive, and False Negative, respectively. Each emotion classification has been evaluated using the parameters listed above. The Authors used the macro-averaged method to generate overall metrics for the six emotion labels. The following are the three-macro metrics:

$$macro - precision = \frac{1}{6}\sum_{i=1}^{6} Pre \tag{46}$$

$$macro - recall = \frac{1}{6}\sum_{i=1}^{6} Rec \tag{47}$$

$$macro - f_1 = \frac{1}{6}\sum_{i=1}^{6} f_1 \tag{48}$$

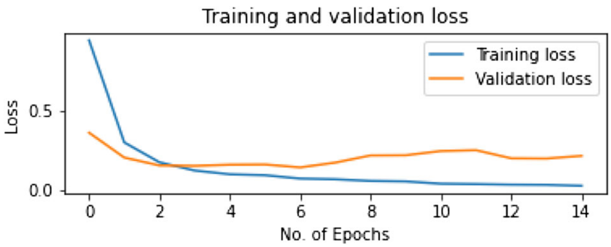


Fig. 4 Training and validation loss of Improved GloVe-HeBiLSTM model

5 Analysis and comparison of experimental results

This section discussed the performance evaluation of Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models. The quality of both models has been assessed using a variety of performance measures, including precision, recall, and F_1 Score. The efficiency of both models has been tested using a confusion matrix. The performance of Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have been compared with existing HeBiLSTM and other baseline models.

5.1 Performance evaluation of improved GloVe-HeBiLSTM model

The Improved GloVe-HeBiLSTM model has been trained and validated on CPU, TPU, and GPU processing units and has used the same parameter settings as used by the proposed HeBi-CuDNNLSTM model. Better results of Improved GloVe-HeBiLSTM has observed on GPU than on CPU and TPU. The execution time taken on GPU is approx 344.23 seconds. The 3.57% training and 26.48% validation loss of the Improved GloVe-HeBiLSTM model in 15 epochs as shown in Fig. 4.

Similarly, the 98.62% training and 93.70% validation accuracy of the Improved GloVe-HeBiLSTM model in 15 epochs as shown in Fig. 5.

The confusion matrix of the Improved GloVe-HeBiLSTM model is illustrated in Fig. 6. The ROC (Receiver Operating Characteristics) curve has been utilized to depict the effectiveness of multi-class categorization problems throughout emotion classification. The ROC curve [54] is a statistical curve that describes the suggested classifier’s ability to differentiate among different classes. The True Positive Rate (TPR) or recall or sensitivity and sensitivity or False Positive Rate defines the ROC curve as expressed in Eqs. (43) and (50) respectively.

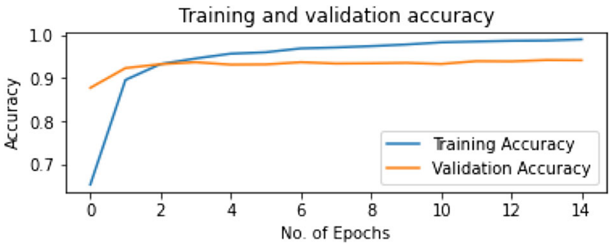


Fig. 5 Training and validation accuracy of Improved GloVe-HeBiLSTM model

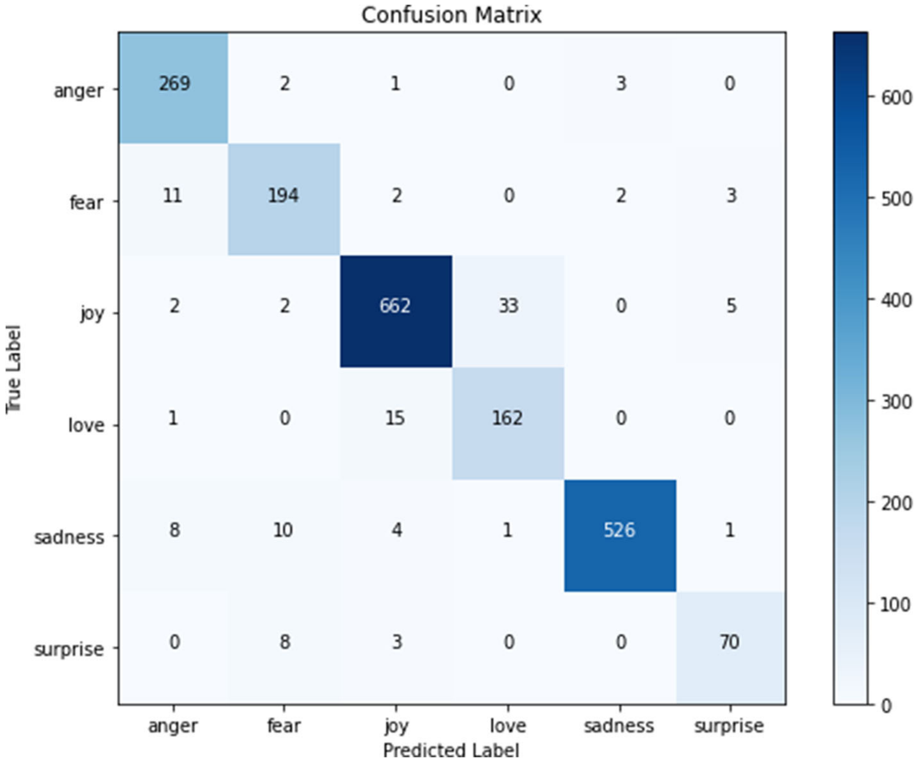


Fig. 6 Confusion matrix of the Improved GloVe-HeBiLSTM model

- **Specificity:**

$$Specificity = TN/(TN + FP) \tag{49}$$

- **False Positive Rate (FPR):**

$$FPR = 1 - Specificity \tag{50}$$

The ROC curve of the Improved Glove-HeBiLSTM model is presented in 7. The Area Under Curve (AUC) is approximately 0.92.

The micro, macro, weighted, and average sample precision of the Improved GloVe-HeBiLSTM model is 94%, 93%, 94%, and 94%, respectively. Similarly, the micro, macro, weighted, and samples average recall value of the Improved GloVe-HeBiLSTM model is 94%, 92%, 94%, and 94%, respectively. And the micro, macro, and weighted, samples average F_1-Score of the Improved GloVe-HeBiLSTM model are 94%, 92%, 94%, and 94% respectively. Therefore, the overall Precision (P), Recall (R), and F_1-Score of the Improved GloVe-HeBiLSTM model for six different emotions are better than existing HeBiLSTM [30] and other recent baseline models as shown in Table 2, 3, and 4 respectively.

5.2 Performance evaluation of proposed HeBi-CuDNNLSTM framework

The Proposed HeBi-CuDNNLSTM framework has been trained and validated on GPU with various parameter settings as mentioned in Table 1. The execution time taken on GPU by

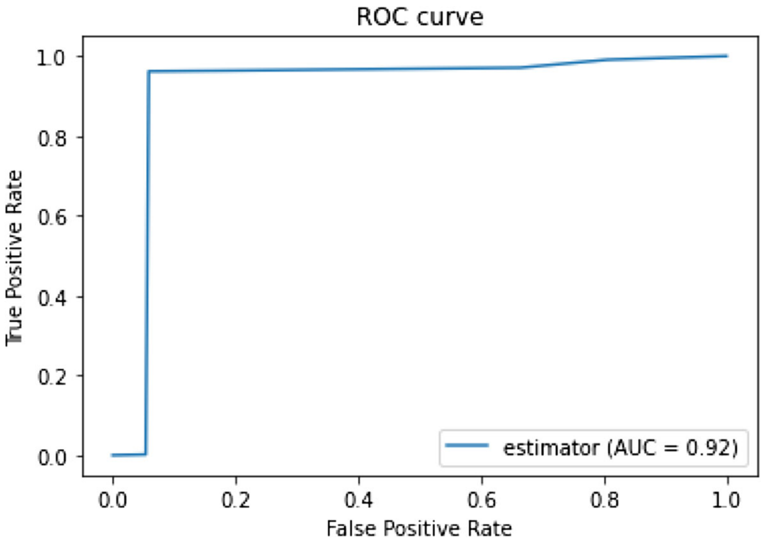


Fig. 7 ROC curve of the Improved GloVe-HeBiLSTM model

the proposed framework was approx 152 seconds. The training and validation loss of the proposed framework is 1.60% and 27.40% respectively in 15 epochs, as shown in Fig. 8.

Similarly, the 99.44% training and 94.20% validation accuracy of the proposed HeBi-CuDNNLSTM model in 15 epochs as shown in Fig. 9.

The micro, macro, weighted, and average sample precision of the proposed HeBi-CuDNNLSTM framework is 95%, 93%, 95%, and 95%, respectively. Similarly, the micro, macro, weighted, and samples average recall value of the proposed framework are 95%, 93%, 95%, and 95%, respectively. The micro, macro, weighted, and sample average F_1-Score of the proposed model are 95%, 93%, 95%, and 95%, respectively, as shown in Table 3. The overall precision (P), recall (R), and F_1-Score of the proposed HeBi-CuDNNLSTM model for six different emotions such as anger, fear, joy, love, sadness, and surprise has shown in Table 4.

Table 2 Comparison of loss and accuracy of Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models with existing HeBiLSTM and other GloVe-based baseline models

Deep Learning Models	Training Loss (%)	Validation Loss (%)	Training Accuracy (%)	Validation Accuracy (%)	Execution Time (In second)
HeBiLSTM [30]	1.46	13.81	97.24	89.00	172.22s
GloVe-CNN-LSTM [20, 52]	7.31	24.63	97.39	92.75	87.82s
GloVe-BiLSTM [35]	2.79	25.38	98.88	93.10	281.27s
GloVe-CNN-BiLSTM [23, 31]	5.49	21.96	97.93	92.65	110.33s
GloVe-CNN-BiGRU [18]	5.60	24.20	97.86	93.25	43.81s
Improved GloVe-HeBiLSTM	**3.57**	**26.48**	**98.62**	**93.70**	**344.23s**
Proposed HeBi-CuDNNLSTM Framework	**1.60**	**27.40**	**99.44**	**94.20**	**152s**

Table 3 Performance comparison of the micro and macro average of the proposed framework with other baseline deep learning models

Deep Learning Models	Micro Average			Macro Average			Weighted Average			Samples Average		
	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)
HeBiLSTM [30]	90	88	89	87	83	85	90	88	89	88	88	88
GloVe-CNN-LSTM [52]	93	93	93	91	90	90	94	93	93	93	93	93
GloVe-BiLSTM [13, 35]	94	94	94	91	91	91	94	94	94	94	94	94
GloVe-CNN-BiLSTM [27, 31]	94	93	93	91	90	90	94	93	93	93	93	93
GloVe-CNN-BiGRU [18]	94	93	94	92	90	91	94	93	94	93	93	93
Improved GloVe-HeBiLSTM	**94**	**94**	**94**	**93**	**92**	**92**	**94**	**94**	**94**	**94**	**94**	**94**
Proposed HeBi-CuDNNLSTM Framework	**95**	**95**	**95**	**93**	**93**	**93**	**95**	**95**	**95**	**95**	**95**	**95**

Table 4 Performance comparison of the proposed framework with other baseline deep learning models

Deep Learning Models	Anger			Fear			Joy			Love			Sadness			Surprise		
	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)
HeBiLSTM [30]	74	55	63	71	58	63	80	63	70	60	50	55	76	**63**	69	67	63	69
GloVe-CNN-LSTM [52]	95	92	**94**	89	84	87	**97**	93	95	83	**97**	**89**	95	**97**	96	85	77	81
GloVe-BiLSTM [13, 35]	**96**	90	93	87	91	89	96	94	95	83	90	87	97	96	96	86	89	87
GloVe-CNN-BiLSTM [27, 31]	93	93	93	91	86	88	96	93	95	86	92	**89**	96	**97**	96	80	86	83
GloVe-CNN-BiGRU [18]	92	**94**	93	**92**	86	89	95	**96**	95	88	87	88	97	96	96	85	79	82
Improved GloVe-HeBiLSTM	**96**	91	93	90	90	**90**	95	**96**	95	**89**	84	86	96	**97**	96	85	**91**	**88**
proposed HeBi-CuDNNLSTM Framework	95	93	**94**	89	**93**	**90**	96	**96**	**96**	87	90	**89**	**98**	**97**	**97**	**95**	78	86

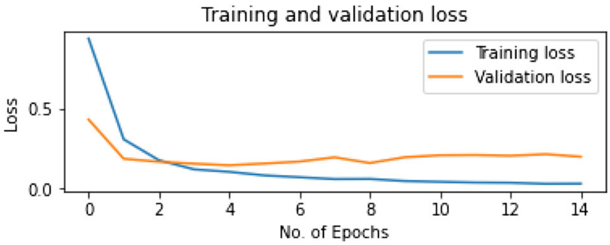


Fig. 8 Training and validation loss of proposed HeBi-CuDNNLSTM framework

The confusion matrix of the proposed HeBi-CuDNNLSTM model has been produced for six different emotions: love, anger, fear, joy, sadness, and surprise. The values for the True label were plotted against the Predicted label as illustrated in Fig. 10. The performance of the proposed HeBi-CuDNNLSTM model for six different emotions is better than the existing HeBiLSTM [30] and other GloVe-based baseline models.
The ROC curve of the proposed model is presented in Fig. 11. The Area Under Curve (AUC) is approximately 0.95.

5.3 Performance comparison

The GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, and Improved GloVe-HeBiLSTM are trained and validated on CPU, TPU, and GPU processing units. The GloVe-CNN-BiGRU and proposed HeBi-CuDNNLSTM framework have been trained and validated on GPU. Authors observed better performance accuracy and execution time of all experimental GloVe-based models on GPU as illustrated in Table 2. The training loss of HeBiLSTM, GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, GloVe-CNN-BiGRU, Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM model is 1.46%, 7.31%, 2.79%, 5.49%, 5.60%, 3.57%, and 1.60% respectively. And the validation loss of HeBiLSTM, GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM,mGloVe-CNN-BiGRU, Improved GloVe-HeBiLSTM, and proposed HeBi-CuDNNLSTM model is 13.81%, 24.63%, 25.38%, 21.96%, 24.20%, 26.48%, and 27.40% respectively as shown in Fig. 12.
Similarly the training accuracy of HeBiLSTM, GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, GloVe-CNN-BiGRU, Improved GloVe-HeBiLSTM and Proposed HeBi-CuDNNLSTM model is 97.24%, 97.39%, 98.88%, 97.93%, 97.86%, 98.62%, and 99.44% respectively. And the validation accuracy of HeBiLSTM, GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, GloVe-CNN-BiGRU, Improved GloVe-HeBiLSTM and

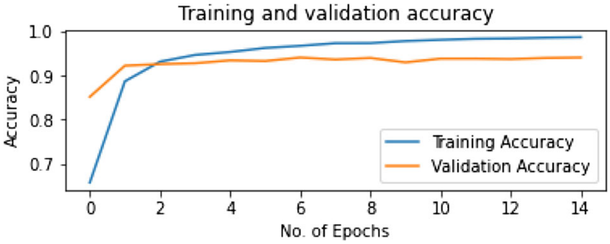


Fig. 9 Training and validation accuracy of proposed HeBi-CuDNNLSTM model

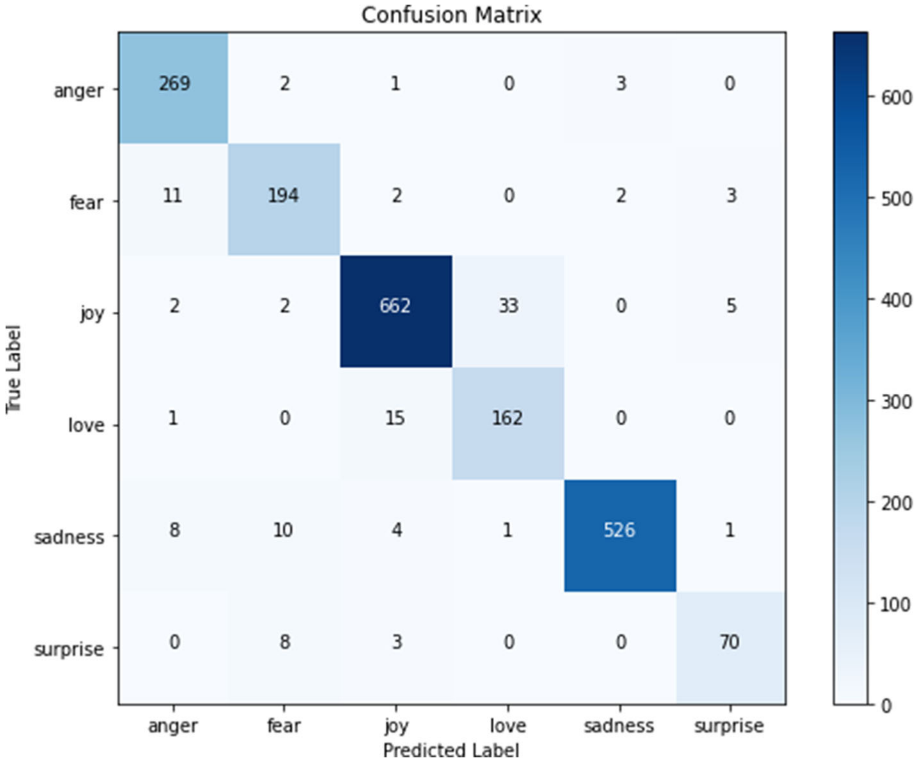


Fig. 10 Confusion matrix of suggested HeBi-CuDNNLSTM framework

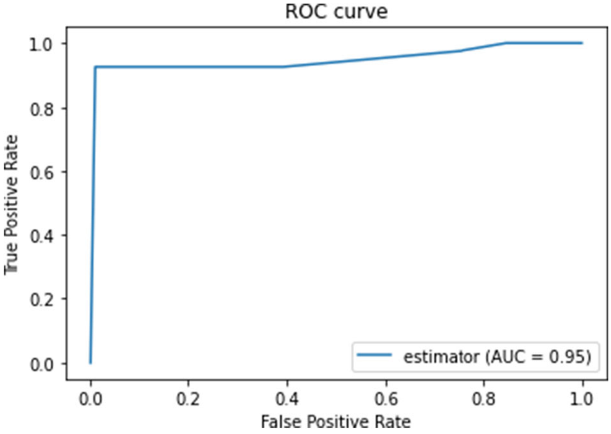


Fig. 11 ROC curve of proposed HeBi-CuDNNLSTM model

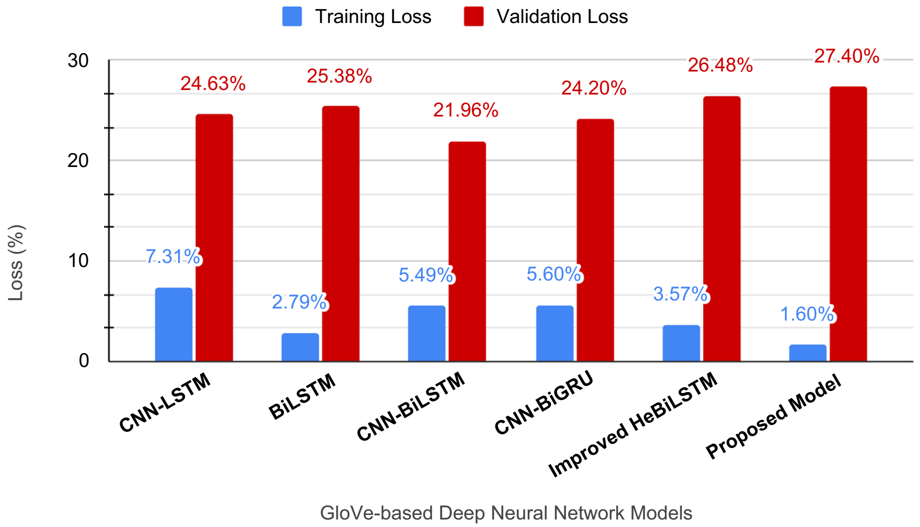


Fig. 12 Comparison of training and validation loss of GloVe-based deep learning models

Proposed HeBi-CuDNNLSTM model is 89%, 92.75%, 93.10%, 92.65%, 93.25%, 93.70%, and 94.20% respectively as shown in Fig. 13.

The approx execution time of HeBiLSTM, GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, GloVe-CNN-BiGRU, Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM model on GPU is 172.22, 87.82, 281.27, 110.33, 43.81, 344.23, and 152 seconds as mentioned in Table 2. Therefore the accuracy of the Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models are higher than the existing HeBiLSTM [30], TF-IDF algorithm [44], and other GloVe-based baseline models.

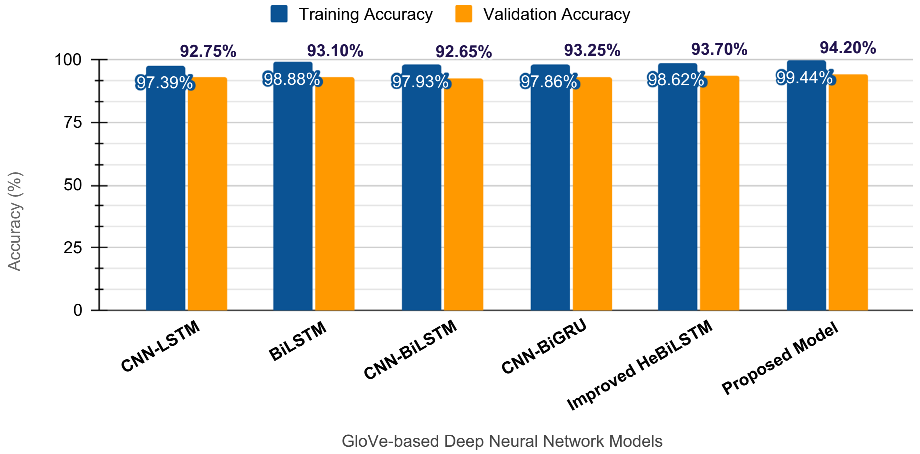


Fig. 13 Comparison of training and validation accuracy of GloVe-based deep learning models

6 Conclusion

This research article enhanced the performance accuracy of the existing HeBiLSTM model named as Improved GloVe-HeBiLSTM model and also proposed a HeBi-CuDNNLSTM model for analyzing six different emotions of textual data. Both models use pre-trained GloVe word vectorization to generate word vectors of numerical values for each word. The Improved GloVe-HeBiLSTM and proposed model have been trained and tested with the *emotions-dataset-for-NLP* dataset available on Kaggle. The effectiveness of Improved GloVe-HeBiLSTM and proposed HeBi-CuDNNLSTM models have been compared with the existing HeBiLSTM, and recent GloVe-based baseline models, including GloVe-CNN-LSTM, GloVe-BiLSTM, GloVe-CNN-BiLSTM, and GloVe-CNN-BiGRU. The performance accuracy of the Improved GloVe-HeBiLSTM (93.70%) and proposed HeBi-CuDNNLSTM (94.20%) models outperform the existing HeBiLSTM [30], TF-IDF algorithm [44], and other baseline models.

In future work, the Authors will implement other hybrid enhanced deep neural networks for multilevel emotion analysis tasks and for analyzing different textual emotions in Hindi, Bengali, and other languages.

Data availability The dataset is taken from Kaggle online digital libraries. The Authors have included the link also in the manuscript.

Declarations

Conflict of interest The Authors declare that they have no conflicts of interest.

References

1. Abid F, Li C, Alam M (2020) Multi-source social media data sentiment analysis using bidirectional recurrent convolutional neural networks. Comput Commun 157:102–115
2. Ahmad Z, Jindal R, Ekbal A, Bhattachharyya P (2020) Borrow from rich cousin: transfer learning for emotion detection using cross lingual embedding. Expert Syst Appl 139:112851
3. Alotaibi FM, Asghar MZ, Ahmad S (2021) A hybrid CNN-LSTM model for psychopathic class detection from Tweeter users. Cogn Comput 13(3):709–723
4. Barbu E (2015) Property type distribution in Wordnet, Corpora and Wikipedia. Expert Syst Appl 42(7):3501–3507
5. Batbaatar E, Li M, Ryu KH (2019) Semantic-emotion neural network for emotion recognition from text. IEEE Access 7:111866–111878
6. Bengio Y, Ducharme R, Vincent P (2000) A neural probabilistic language model. Adv Neural Inf Proces Syst 13
7. Bhatti UA, Huang M, Wang H, Zhang Y, Mehmood A, Di W (2018) Recommendation system for immunization coverage and monitoring. Hum Vaccin Immunother 14(1):165–171
8. Bhatti UA, Huang M, Wu D, Zhang Y, Mehmood A, Han H (2019) Recommendation system using feature extraction and pattern recognition in clinical care systems. Enterp Inf Syst 13(3):329–351
9. Bhatti UA, Yu Z, Chanussot J, Zeeshan Z, Yuan L, Luo W, Nawaz SA, Bhatti MA, Ain QU, Mehmood A (2021) Local similarity-based spatial-spectral fusion hyperspectral image classification with deep CNN and Gabor filtering. IEEE Trans Geosci Remote Sens 60:1–15
10. Bhatti UA, Yu Z, Li J, Nawaz SA, Mehmood A, Zhang K, Yuan L (2020) Hybrid watermarking algorithm using clifford Algebra with Arnold Scrambling and chaotic encryption. IEEE Access 8:76386–76398
11. Bhatti UA, Zeeshan Z, Nizamani MM, Bazai S, Yu Z, Yuan L (2022) Assessing the change of ambient air quality patterns in Jiangsu Province of China pre-to post-COVID-19. Chemosphere 288:132569
12. Chatterjee A, Narahari KN, Joshi M, Agrawal P (2019) SemEval-2019 task 3: Emocontext contextual emotion detection in text. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp 39–48

13. Chen K, Mahfoud RJ, Sun Y, Nan D, Wang K, Haes Alhelou H, Siano P (2020) Defect texts mining of secondary device in smart substation with GloVE and attention-based bidirectional LSTM. Energies 13(17):4522
14. Chen M, Hao Y, Hwang K, Wang L, Wang L (2017) Disease prediction by machine learning over big data from healthcare communities. IEEE Access 5:8869–8879
15. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. Preprint at http://arxiv.org/abs/1810.04805
16. Faruqui M, Dodge J, Jauhar SK, Dyer C, Hovy E, Smith NA (2014) Retrofitting word vectors to semantic lexicons. Preprint at http://arxiv.org/abs/1411.4166
17. François C et al (2015) Keras: The python deep learning library. Keras. IO
18. Gao Z, Li Z, Luo J, Li X (2022) Short text aspect-based sentiment analysis based on CNN+ BiGRU. Appl Sci 12(5):2707
19. Gers FA, Schmidhuber E (2001) LSTM recurrent networks learn simple context-free and context-sensitive languages. IEEE Trans Neural Networks 12(6):1333–1340
20. Hegde SU, Zaiba A, Nagaraju Y et al (2021) Hybrid CNN-LSTM model with GloVE word vector for sentiment analysis on football specific tweets. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). IEEE, pp 1–8
21. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. Preprint at http://arxiv.org/abs/1207.0580
22. Jayakrishnan R, Gopal GN, Santhikrishna M (2018) Multi-class emotion detection and annotation in Malayalam novels. In: 2018 International Conference on Computer Communication and Informatics (ICCCI). IEEE, pp 1–5
23. Kamyab M, Liu G, Adjeisah M (2021) Attention-based CNN and BI-LSTM model based on TF-IDF and GloVE word embedding for sentiment analysis. Appl Sci 11(23):11255
24. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. Preprint at http://arxiv.org/abs/1412.6980
25. Kratzwald B, Ilić S, Kraus M, Feuerriegel S, Prendinger H (2018) Deep learning for affective computing: text-based emotion recognition in decision support. Decis Support Syst 115:24–35
26. Kumar P, Babulal KS, Mahto D, Khurshid Z (2023) Analyzing deep neural network algorithms for recognition of emotions using textual data. In: Key Digital Trends Shaping the Future of Information and Management Science: Proceedings of 5th International Conference on Information Systems and Management Science (ISMS) 2022. Springer, pp 60–70
27. Li A, Yi S (2022) Emotion analysis model of microblog comment text based on CNN-BiLSTM. Comput Intell Neurosci 2022
28. Li W, Li Y, Liu W, Wang C (2022) An influence maximization method based on crowd emotion under an emotion-based attribute social network. Inf Process Manag 59(2):102818
29. Ma L, Zhang L, Ye W, Hu W (2019) PKUSE at SemEval-2019 task 3: emotion detection with emotion-oriented neural attention network. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp 287–291
30. Mahto D, Yadav SC (2022) Hierarchical Bi-LSTM based emotion analysis of textual data. Bull Pol Acad Sci Tech 70:e141001–e141001
31. Mahto D, Yadav SC, Lalotra GS (2022) Sentiment prediction of textual data using hybrid convbidirectional-LSTM model. Mob Inf Syst 2022
32. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. Preprint at http://arxiv.org/abs/1301.3781
33. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Proces Syst 26
34. Nandanwar AK, Choudhary J (2021) Semantic features with contextual knowledge-based web page categorization using the GloVE model and stacked BiLSTM. Symmetry 13(10):1772
35. Pei Y, Chen S, Ke Z, Silamu W, Guo Q (2022) AB-LaBSE: Uyghur sentiment analysis via the pre-training model with BiLSTM. Appl Sci 12(3):1182
36. Pennington J, Socher R, Manning CD (2014) GloVE: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp 1532–1543
37. Pimpalkar A et al (2022) MBiLSTMGloVE: embedding GloVE knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis. Expert Syst Appl 203:117581
38. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9
39. Ragheb W, Azé J, Bringay S, Servajean M (2019) Attention-based modeling for emotion detection and classification in textual conversations. Preprint at http://arxiv.org/abs/1906.07020

40. Ramachandran P, Zoph B, Le QV (2017) Searching for activation functions. Preprint at http://arxiv.org/abs/1710.05941
41. Rani S, Bashir AK, Alhudhaif A, Koundal D, Gunduz ES et al (2022) An efficient CNN-LSTM model for sentiment detection in# blacklivesmatter. Expert Syst Appl 193:116256
42. Rao G, Huang W, Feng Z, Cong Q (2018) LSTM with sentence representations for document-level sentiment classification. Neurocomputing 308:49–57
43. Serrano-Guerrero J, Bani-Doumi M, Romero FP, Olivas JA (2022) Understanding what patients think about hospitals: a deep learning approach for detecting emotions in patient opinions. Artif Intell Med 128:102298
44. Sundaram V, Ahmed S, Muqtadeer SA, Reddy RR (2021) Emotion analysis in text using TF-IDF. In: 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, pp 292–297
45. Tam S, Said RB, Tanriöver ÖÖ (2021) A ConvBiLSTM deep learning model-based approach for twitter sentiment classification. IEEE Access 9:41283–41293
46. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp 384–394
47. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Proces Syst 30
48. Wang J, Peng B, Zhang X (2018) Using a stacked residual LSTM model for sentiment intensity prediction. Neurocomputing 322:93–101
49. Wu J-L, He Y, Yu L-C, Lai KR (2020) Identifying emotion labels from psychiatric social texts using a bi-directional LSTM-CNN model. IEEE Access 8:66638–66646
50. Yu L-C, Wang J, Lai KR, Zhang X (2017) Refining word embeddings using intensity scores for sentiment analysis. IEEE/ACM Trans Audio Speech Lang Process 26(3):671–681
51. Yu L-C, Wang J, Lai KR, Zhang X (2018) Pipelined neural networks for phrase-level sentiment intensity prediction. IEEE Trans Affect Comput 11(3):447–458
52. Zhang W, Li L, Zhu Y, Yu P, Wen J (2022) CNN-LSTM neural network model for fine-grained negative emotion computing in emergencies. Alex Eng J 61(9):6755–6767
53. Zhao J, Lin J, Liang S, Wang M (2021) Sentimental prediction model of personality based on CNN-LSTM in a social media environment. J Intell Fuzzy Syst 40(2):3097–3106
54. Zhou J, Lu Y, Dai H-N, Wang H, Xiao H (2019) Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM. IEEE Access 7:38856–38866

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.