Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.035352
Article

Text Sentiment Analysis Based on Multi-Layer Bi-Directional LSTM with a Trapezoidal Structure

Zhengfang He[1,2,*], Cristina E. Dumdumaya[2] and Ivy Kim D. Machica[2]

[1]School of Intelligent Science and Engineering, Yunnan Technology and Business University, Kunming, 650000, China
[2]College of Information and Computing, University of Southeastern Philippines, Davao City, Davao del Sur, Philippines
*Corresponding Author: Zhengfang He. Email: zhengfang_he@usep.edu.ph
Received: 17 August 2022; Accepted: 13 January 2023

Abstract: Sentiment analysis, commonly called opinion mining or emotion artificial intelligence (AI), employs biometrics, computational linguistics, natural language processing, and text analysis to systematically identify, extract, measure, and investigate affective states and subjective data. Sentiment analysis algorithms include emotion lexicon, traditional machine learning, and deep learning. In the text sentiment analysis algorithm based on a neural network, multi-layer Bi-directional long short-term memory (LSTM) is widely used, but the parameter amount of this model is too huge. Hence, this paper proposes a Bi-directional LSTM with a trapezoidal structure model. The design of the trapezoidal structure is derived from classic neural networks, such as LeNet-5 and AlexNet. These classic models have trapezoidal-like structures, and these structures have achieved success in the field of deep learning. There are two benefits to using the Bi-directional LSTM with a trapezoidal structure. One is that compared with the single-layer configuration, using the of the multi-layer structure can better extract the high-dimensional features of the text. Another is that using the trapezoidal structure can reduce the model's parameters. This paper introduces the Bi-directional LSTM with a trapezoidal structure model in detail and uses Stanford sentiment treebank 2 (STS-2) for experiments. It can be seen from the experimental results that the trapezoidal structure model and the normal structure model have similar performances. However, the trapezoidal structure model parameters are 35.75% less than the normal structure model.

Keywords: Text sentiment; Bi-directional LSTM; Trapezoidal structure

1 Introduction

Sentiment analysis (SA), commonly referred to as opinion mining, examines, interprets, and summarizes pertinent texts containing many emotions. It belongs to the field of natural language processing (NLP) [1]. Its purpose is to swiftly and effectively extract the emotional content (both positive and negative) that a text expresses through particular techniques [2]. Sentiment analysis is essential in scientific research and has a wide range of practical applications, according to its application value and field [3–5]. As a result, many scholars began to focus on using the sentimental information provided by netizens to understand how the public feels about various products, occasions, or personalities [6].

For instance, people may find convenience in sentiment analysis techniques and methodologies. When a hot topic or contentious event arises in a society, the government can compile online comments and opinions from the general population as part of its intelligence surveillance program. The corpus above is investigated using sentiment analysis-related technologies to determine and monitor the most popular public opinion inclination. It does this by utilizing the legitimacy and power of the government to influence society's proper value orientation, maintain social order, and calm internet users' emotions when situations occur [7–9]. Further, any group's collective opinions and emotional attitudes are significant in this era, and sentiment analysis is utilized to harness its value successfully. Hence, research and development of appropriate theories and practices are crucial [10,11].

In recent years, online social networking has grown significantly [12], the scale and dimension of textual data are rapidly increasing [13]. Drawing reliable conclusions from complex public opinion and online evaluation through human learning takes much work. Therefore, many sentiment analysis algorithms have been developed and enhanced to solve this problem. There are three types of sentiment and emotion analysis techniques: emotion lexicon, traditional machine learning, and deep learning [14].

Compared to emotion lexicon and traditional machine learning algorithms. The deep learning algorithms can capture more comprehensive text features due to the sophistication of their models when faced with massive data. A better understanding of the text can also achieve better results in sentiment analysis. Therefore, the methods using deep learning have become the mainstream methods.

In the text sentiment analysis algorithm based on deep learning, multi-layer Bi-directional LSTM is widely used [15–18], but the parameters of these models are too huge. And the training and inference phases of this model require more CPU and memory resources. So, this paper proposes the Bi-directional LSTM with a trapezoidal structure model. There are two benefits to using the Bi-directional LSTM with a trapezoidal structure.

1. One is that compared with the single-layer structure, the use of the multi-layer structure can better extract the high-dimensional features of the text.
2. Another is that using the trapezoidal structure can reduce the model's parameters.

Reducing model parameters can be beneficial in some contexts, such as edge computing and the Internet of things (IoT) [19], because fewer parameters require lesser processing resources to enable inference on these devices than a typical structural model would.

2 Literature Review

2.1 Emotion Lexicon based Algorithms

The notion that documents are made up of related words in the text has changed due to sentiment analysis based on a lexicon. According to a given probability distribution, the text is viewed as a collection of phrases or words, and all words expressing feeling are matched with the authoritative lexicon. The calculating outcome of the matching comparison explains the text's sentimental polarity.

Generally, algorithms based on an emotion lexicon need to build a lexicon and assign different sentiment labels or scores to the words in the lexicon. The earliest sentiment lexicon acknowledged by the academic community, general inquirer (GI), had a capacity of more than 4,200 sentiment words, 1,915 positive words and 2,291 negative words. After that, larger-capacity sentimental lexicons became more prevalent. For instance, a sentiment lexicon called Opinion Lexicon was created by Hu et al. with a capacity of around 6,800 words, 2,006 positive words and 4,783 negative words [20]. Following that, the SentiWordNet was proposed by Esuli et al. [21] a lexical resource in which each synset of WordNet was associated with three numerical scores Obj (s), Pos (s), and Neg (s), describing how Objective, Positive, and Negative the terms contained in the synset were. Each of the three scores ranges from 0.0 to 1.0, and their sum was 1.0 for each synset. This means that a synset might have nonzero scores for all the three categories, each of the three opinion-related properties only to a certain degree. The latest version was SentiWordNet 3.0 [22].

Only using an emotion lexicon with positive and negative for sentiment analysis is ineffective, so there is much work to research on improving the performance. Turney [23] offered a simple unsupervised learning algorithm for categorizing reviews. The algorithm extracted phrases containing adjectives or adverbs, estimated each phrase's semantic orientation by pointwise mutual information and information retrieval (PMI-IR), then classified the review based on the average semantic orientation of the phrases. A system was developed by Kim et al. [24]; it automatically extracted the pros and cons from online reviews by combining lexical features, positional features, and opinion-bearing word features in sentences. Taboada et al. [25] presented a polarity classification method based on emotion lexicon, the semantic orientation CALculator (SO-CAL) used lexicons of words annotated with their semantic orientation (polarity and strength), and incorporated intensification and negation.

The precise operation is to compare and tally with the authoritative sentiment lexicon and compute each corpus' score as the text's sentiment score following predetermined guidelines. The numerical component represents the intensity of sentiments, and the positive and negative signs represent the positive and negative feelings, respectively. Counting and statistics are inseparable from the rule lexicon's sentiment analysis. Natural language, however, contains a wealth of knowledge. Coherent sentences convey much information, whereas isolated words may not have much meaning. Additionally, the impact of context on sentimental information is more significant. As a result, mechanized word segmentation and counting may be limited and have faulty semantic comprehension.

2.2 Traditional Machine Learning based Algorithms

The core idea of the sentiment analysis model based on machine learning is to digitize textual data and convert it into a form that machines can recognize. The artificially labeled raw data is divided into training and test sets, then the model with specific parameters obtained under the training set is used to achieve sentiment classification. In this field, the mainstream models are naive Bayes (NB), maximum entropy (ME), and support vector machine (SVM).

The NB model for sentiment analysis has been the subject of numerous investigations. Govindarajan, for instance, discovered multiple NB and genetic algorithm combinations. It was found that the classifiers combined with NB and genetic algorithms had a considerable classification effect [26]. Eronen used the NB model to analyze public opinion while crawling text data from news websites [27]. Wikarsa et al. discovered by literature research that few NB models were used for sentiment analysis of corpora on Twitter, but these experiments obtained a high accuracy [28]. Similarly, Dey et al. utilized the NB algorithm to study the sentiment classification of review corpora (such as movies and hotels) and found moderate results [29].

ME has also been successfully applied in numerous NLP applications. Berger et al. proposed the maxim likelihood method to generate ME models automatically as a demonstration [30]. To address the issue that words from the lexicon occasionally failed to convey emotional inclinations in particular settings, Fei et al. devised an approach based on the ME classification model to identify emotional terms in comments [31]. Likewise, Batista et al. proposed a binary maximum entropy classifier for automatic sentiment analysis and topic classification of Spanish Twitter data [32].

Sharma et al. [33] gathered a corpus of 2000 reviews and suggested a BoostedSVM model built on SVM through algorithmic development. The experimental findings show that the modified model was superior to the initial SVM model. In [34], metaphors on Twitter were employed as the research object by Karanasou et al. based on the SVM model. The classification effect of SVM on the linear kernel function was improved by calculating the similarity, and it was found that the enhanced model had a better effect on the metaphor. Han et al. [35] illustrated a Fisher kernel SVM (FK-SVM) function based on probabilistic latent semantic analysis for sentiment analysis by SVM. The experimental results showed that the FK-SVM model was better than the basic SVM models.

The training effect of traditional machine learning models usually depends on the quality of data preprocessing. The training outcome is typically a locally optimal solution because it involves human interaction. There is no set norm, and there are many uncertainties. The randomness of numerical vectors makes traditional machine learning models for sentiment analysis less robust.

2.3 Deep Learning based Algorithms

Due to the powerful learning ability of deep learning, it is widely utilized in various fields and has achieved major breakthroughs. For instance, Bengio et al. [36] proposed an n-gram language model consisting of a neural network with an input layer, a hidden layer, and an output layer. Training the model to provide a distributed representation of words avoided the dimensional disaster catastrophe an enormous lexicon brings. It also benefits from semantic similarity. Subsequently, Hinton et al. published articles on language models [37–39].

Word vector technique is credited with helping deep learning in NLP to evolve. Mikolov et al. created a word2vec model to vectorize words and mine word similarity through vectors. An interesting finding in this work was additive compositionality. This was crucial for natural language processing tasks [40,41]. Also, Pennington et al. proposed the glove model, which used local and contextual information to obtain word vector representations during training [42].

The development of word vectors has led researchers to use deep learning for sentiment analysis. Kim used word vectors to represent text and convolutional neural networks to extract phrase representations, outperforming traditional machine learning in sentiment classification [43]. For text sentiment analysis, Khatri et al. built a single neural network model that produced promising experimental outcomes. However, the recognition accuracy was reduced for texts with complex emotional features [44]. According to the sentence structure, Wang et al. set up convolution kernels of various lengths, divided the text data into several regions, extracted local features from the text regions using convolution and pooling operations, and then extracted long-distance dependency data using the LSTM model [45].

Meanwhile, Liu et al. trained unlabeled samples to extract features through an unsupervised algorithm and then conducted sentiment analysis through an LSTM model. The results show that combined supervised and unsupervised methods could enhance the model's classification performance [46]. A deep recurrent neural network (RNN) built on Bi-directional LSTM was proposed by Sharfuddin et al. and utilized for sentiment analysis [47]. Xu et al. proposed to use Bi-directional LSTM to extract text features and then performed sentiment analysis through a feedforward neural network classifier [48]. Also, some researchers had applied multi-layer Bi-directional LSTM models to sentiment analysis [15–18]. However, the parameters of these models are too large because the layers of the models have identical numbers of neurons.

Compared with traditional sentiment analysis methods, deep learning has a flexible network structure, can extract abstract nonlinear features using nonlinear functions without additional lexicons and complex feature extraction algorithms, and has significant research value.

3 Multi-Layer Bi-Directional LSTM with a Trapezoidal Structure

3.1 Overall Model

A schematic diagram of the overall model is shown in Fig. 1. The design of the trapezoidal structure is derived from classic neural networks, such as LeNet-5 [49], AlexNet [50]. These classic models have trapezoidal-like structures, and these structures have achieved success in the field of deep learning.

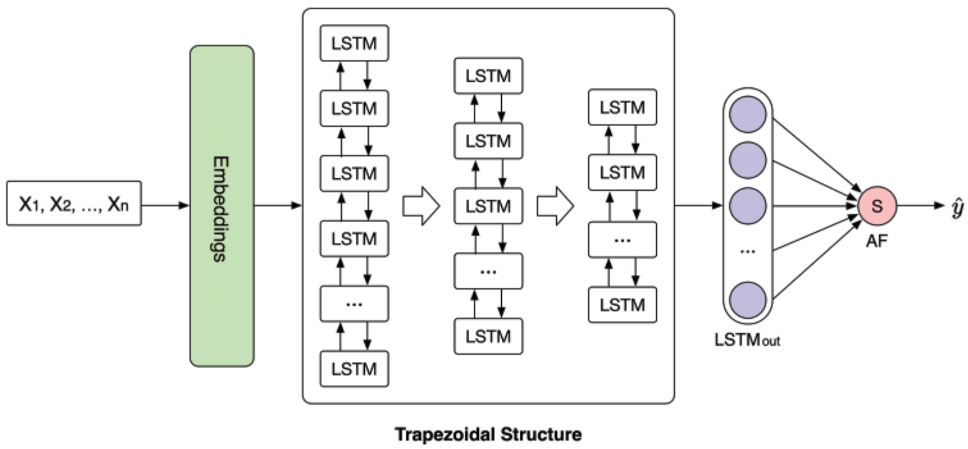


Figure 1: Overall model

The Bi-directional LSTM with a trapezoidal structure has two advantages. One is that the usage of the multi-layer structure can better extract the high-dimensional features of the text as compared to the single-layer structure. Another is that the model's parameters can be lowered by using the trapezoidal structure.

In Fig. 1, the model's input is $x_1, x_2, \ldots, x_n$; the output is $\hat{y}$. First, input the $x_1, x_2, \ldots, x_n$ to the embeddings, and get the words' embeddings vectors. Then input the embeddings vectors to the trapezoidal structure to get $LSTM_{out}$. Finally, the $LSTM_{out}$ is connected by a fully connected layer, and the Sigmoid activation function is used to get the model's output $\hat{y}$.

Some data examples of input and output for the model are shown in Table 1.

As shown in Table 1, when the input is a positive sentence, "*The weather is so nice today.*", the corresponding output of the model is 1. Conversely, when the input is a negative sentence, "*I am saddened by the results of the experiment*," the corresponding output of the model is 0.

Table 1: Data examples of input and output

Input	Output	
	Emoji	Label
The weather is so nice today.	😀	**1**
I am saddened by the results of the experiment.	☹️	**0**

3.2 Embeddings

Sentence sequences are entered into the neural network's input layer. The dimension is $1 \times T$, where T represents the length of the sequence, and each item is the index of the word in the lexicon. Since the input layer accepts a sequence of integers, it is necessary to convert the *index* into a distributed word vector. This paper uses the embeddings layer to accomplish this goal, which converts integers into vectors with fixed dimensions. This layer accepts a $1 \times T$ dimension vector, outputs a $V \times T$ matrix, randomly initializes the weight matrix W_{VN}, and performs two-step operations on the input integer *index*: First, a unit column vector $\vec{x}^{T}$ of $N \times 1$ dimension is generated, where the *index* position is 1; then do a matrix multiplication of W_{VN} and $\vec{x}^{T}$. The result is shown in Eq. (1).

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{V1} & w_{V2} & \cdots & w_{VN} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} w_{1i} \\ w_{2i} \\ \vdots \\ w_{Vi} \end{bmatrix} \tag{1}$$

The i in Eq. (1) is the *index*. The essence of Eq. (1) is a table lookup process, extracting the vector of the *index* column, and each column of the matrix W_{VW} represents the distributed representation of the corresponding lexicon. A neural network layer implements embeddings. This paper uses joint training for embeddings layer.

3.3 The Trapezoidal Structure

This paper's multi-layer Bi-directional LSTM model has a trapezoidal structure, which is strictly an isosceles trapezoid. An isosceles trapezoid is a trapezoid where the base angles have the same measure. Consequently, the two legs are also of equal length and have reflection symmetry. The isosceles trapezoid is shown in Fig. 2.

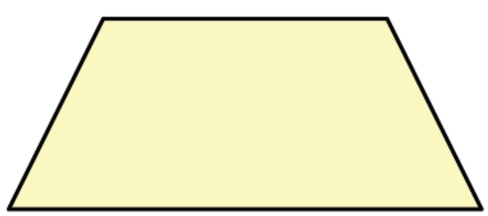


Figure 2: The isosceles trapezoidal

The model designed in this paper is shown in Fig. 3, where h^1 represents the first hidden layer, h^2 represents the second hidden layer, and the sliding window length is K. The sliding window slides from left to right. The average value of the neurons in the sliding window is calculated each time, and a neuron in the next layer is obtained. The number of neurons in h^1 is N. After sliding to the right, a total of $N - K + 1$ neurons in the second layer are generated. In the same way, sliding on the second layer can generate $N-2K+2$ neurons in the third layer. So, a decreasing trapezoid structure is formed.

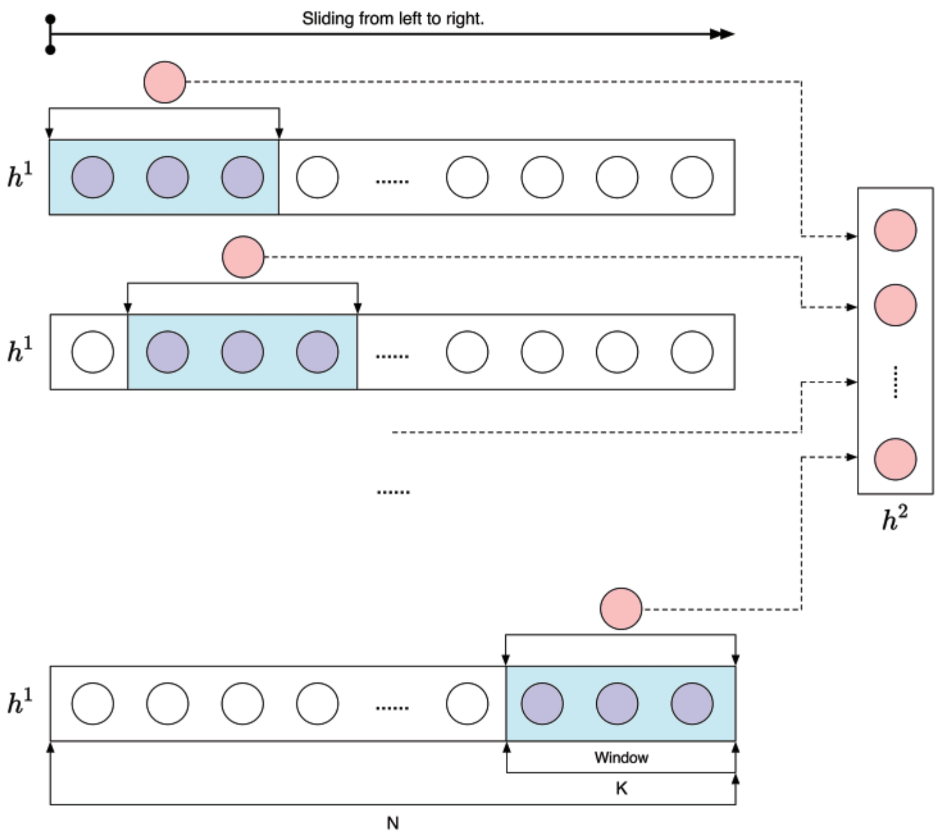


Figure 3: The method of trapezoidal structure

If $N=9$ and $K=3$, the model will be expanded as shown in Fig. 4.

Fig. 3 is a schematic diagram of sliding from left to right, but this paper uses a Bi-directional LSTM. So, the neurons of the first layer are $2N$; $2N-2K+2$ and $2N-4K+4$ are for the second and third layers. The mathematical expressions behind setting up the Bi-directional LSTM hidden layers are shown in Eqs. (2) and (3). Equation (2) represents layer 1, and Eq. (3) represents layers 2 and 3. In Eqs. (2) and (3), the only difference between these two relationships is in the direction of recursion through the corpus.

$$\begin{aligned} \overrightarrow{h}_t^{(i)} &= f\left(\overrightarrow{W}^{(i)} x_t + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)}\right) \\ \overleftarrow{h}_t^{(i)} &= f\left(\overleftarrow{W}^{(i)} x_t + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)}\right) \end{aligned}, \quad where \quad i=1 \quad and \quad t \in [1, N] \tag{2}$$

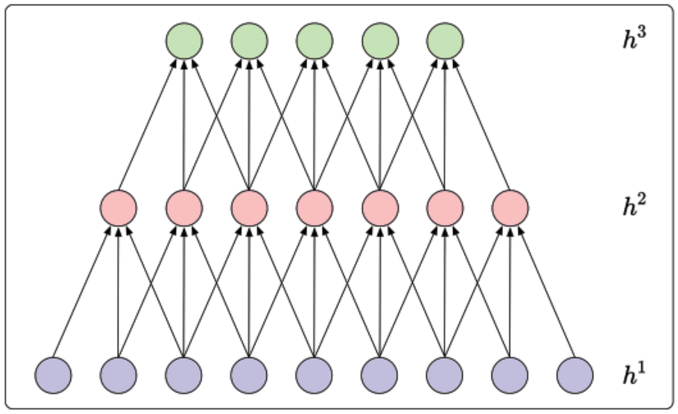


Figure 4: The expanded model

$$\begin{aligned} \overrightarrow{h}_t^{(i)} &= f\left(\overrightarrow{W}^{(i)}\frac{\sum_{j=0}^{K-1} h_{t+j}^{(i-1)}}{K} + \overrightarrow{V}^{(i)}\overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)}\right) \\ \overleftarrow{h}_t^{(i)} &= f\left(\overleftarrow{W}^{(i)}\frac{\sum_{j=0}^{K-1} h_{t+j}^{(i-1)}}{K} + \overleftarrow{V}^{(i)}\overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)}\right) \end{aligned}, \quad where \quad i > 1 \quad and \quad t \in [1,\ N - (i-1) \times (K-1)] \tag{3}$$

In Eq. (3), the average value of the neurons in the sliding window is calculated by $\frac{\sum_{j=0}^{K-1} h_{t+j}^{(i-1)}}{K}$, and a neuron in the next layer is obtained. Because the average value of the sliding window is calculated, the neurons in the hidden layer will decrease with the increase of hidden layers, consequently forming a trapezoidal structure.

In this paper, $N = 128$ and $K = 32$. So, the detailed parameters of the trapezoidal structure are shown in Table 2.

Table 2: The detailed parameters of the trapezoidal structure

Layer	Output shape
Bidirectional_1	(Max_Length, 256)
Bidirectional_2	(Max_Length, 194)
Bidirectional_3	(132)

3.4 Bi-Directional LSTM

The neurons of the trapezoidal structure in Fig. 1 use Bi-directional LSTM. A Bi-directional deep neural network maintains two hidden layers at each time-step, t, one for the left-to-right propagation and another for the right-to-left propagation. This network consumes twice as much memory space for its weight and bias parameters to maintain two hidden layers at any time.

Unlike standard feedforward neural networks, LSTM has feedback connections. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. As shown in Fig. 5, the cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The mathematical formulations of LSTM are shown in Eq. (4).

$$\begin{aligned} & i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) && (\textit{Input gate}) \\ & f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) && (\textit{Forget gate}) \\ & o_t = \sigma\,(W_o x_t + U_o h_{t-1} + b_o) && (\textit{Output gate}) \\ & \tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) && \\ & c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t && (\textit{Cell}) \\ & h_t = o_t \odot tanh\,(c_t) && \end{aligned} \tag{4}$$

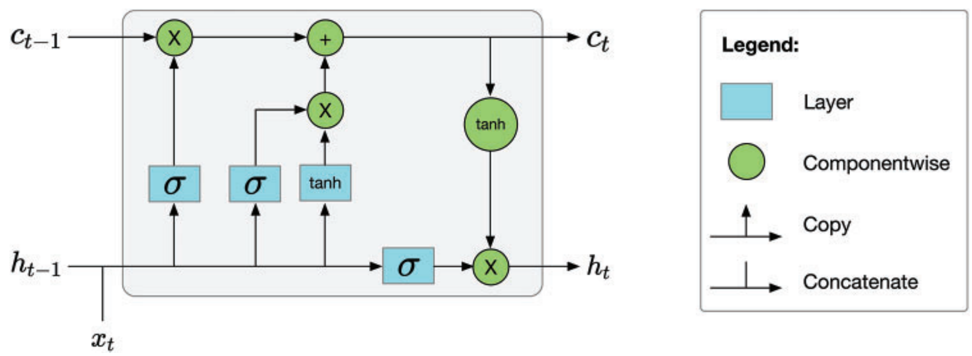


Figure 5: The LSTM cell

As shown in Fig. 5 and Eq. (4), LSTM consists of a cell and three gates. LSTM networks are well-suited for classifying, processing, and making predictions based on time series data. LSTMs were developed to deal with the vanishing gradient problem encountered when training traditional Recurrent Neural Networks (RNNs).

3.5 Training Algorithms of this Model

For the overall model, the input to the model is $x_1, x_2, \ldots, x_n$. First, input the $x_1, x_2, \ldots, x_n$ to the embeddings layer using Eq. (1). Then through the trapezoidal structure with three layers, the first layer uses Eq. (2), and the second and third layers use Eq. (3). The output of trapezoidal structure is obtained by Eq. (5).

$$TS_{out} = g\,(Uh_t + c) = g\left(U\left[\overrightarrow{h}_t; \overleftarrow{h}_t\right] + c\right) \tag{5}$$

In Eq. (5), *TS* represents trapezoidal structure and shows the classification relationship used for predicting the next word via summarizing past and future word representations.

After the TS_{out} is obtained from the trapezoidal structure, all the features are connected through a fully connected layer by Eq. (6).

$$\hat{y} = g\,(W \cdot TS_{out} + b) \tag{6}$$

As shown in Fig. 1, the overall model is a binary classification problem, so g in Eq. (6) is a Sigmoid function [51]. Since g is a Sigmoid function, the entire model can be regarded as a logistic regression model; hence the model's loss function is shown in Eq. (7).

$$L\left(\hat{y}, y\right) = -\left[yln\left(\hat{y}\right) + (1-y)\, ln\left(1-\hat{y}\right)\right] \tag{7}$$

In Eq. (7), $\hat{y}$ is the predicted value and y is the real value. It explicitly represents that the loss function L is parameterized by the weights and biases, which it generally refers to in machine learning as θ ($\theta = w_1, w_2, \cdots, w_j; b_1, b_2, \cdots, b_k$). So, the goal is to find a set of weights and biases that minimizes the loss function, averaged over all samples are shown in Eq. (8).

$$\hat{\theta} = \underset{\theta}{argmin} \frac{1}{m} \sum_{i=1}^{m} L\left(f\left(x^{(i)}; \theta\right), y^{(i)}\right) \tag{8}$$

Then the gradient descent algorithm is used to update θ. For each sample, its update process is shown in Eq. (9).

$$\begin{aligned} &\theta_{t+1} = \theta_t - \eta \nabla L\left(f\left(x; \theta\right), y\right) \\ &where\ \nabla L\left(f\left(x;\theta\right), y\right) = \begin{bmatrix} \frac{\partial}{\partial w_1} L\left(f\left(x;\theta\right), y\right) \\ \frac{\partial}{\partial w_2} L\left(f\left(x;\theta\right), y\right) \\ \vdots \\ \frac{\partial}{\partial w_j} L\left(f\left(x;\theta\right), y\right) \\ \frac{\partial}{\partial b_1} L\left(f\left(x;\theta\right), y\right) \\ \frac{\partial}{\partial b_2} L\left(f\left(x;\theta\right), y\right) \\ \vdots \\ \frac{\partial}{\partial b_k} L\left(f\left(x;\theta\right), y\right) \end{bmatrix} \end{aligned} \tag{9}$$

To update θ more efficiently, this paper uses the backpropagation algorithm to update the parameters. In this model, h^3 is updated first, followed by h^2 and h^1.

4 Experiments and Results Analysis

4.1 Data Set and Data Pre-processing

This paper uses the SST-2 [52] dataset. The dataset consists of sentences from movie reviews and human sentiment annotations. The task is to predict the sentiment of a given sentence. The dataset uses the two-way (positive/negative) class split and uses only sentence-level labels. The number of samples is 67,349 for the training and 1,821 for the test.

4.2 Comparison Experiments

In order to verify the validity of the model proposed in this paper, a set of comparative models is designed for experiments. They are the trapezoidal structure model and the normal structure model, respectively. Their parameters are shown in Table 3.

Table 3: The parameters of the trapezoidal structure model and the normal structure model

Layer	Trapezoidal structure		Normal structure	
	Output shape	Param #	Output shape	Param #
Bidirectional_1	(Max_Length, 256)	263168	(Max_Length, 256)	263168
Bidirectional_2	(Max_Length, 194)	274704	(Max_Length, 256)	394240
Bidirectional_3	(132)	137808	(256)	394240

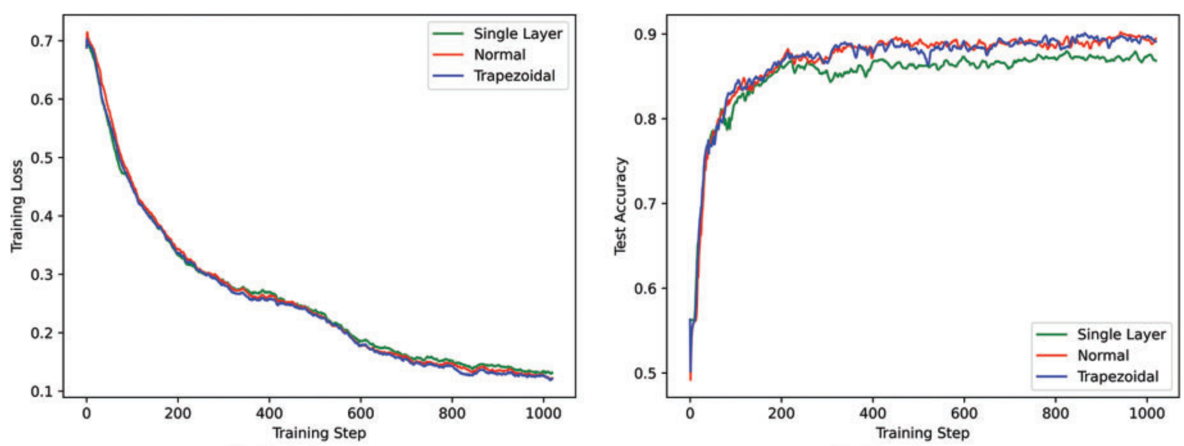
This paper uses TensorFlow [53] coding and experiments on GPU. The main parameters of model training are shown in Table 4.

Table 4: Main parameters of model training

Parameters	Value
max_length	32
batch_size	128
training_step	1020
learning_rate	0.005
optimizer	AdamOptimizer

4.3 Results and Analysis

In order to verify the trapezoidal structure model's effectiveness, this paper adds a single layer Bi-directional LSTM as the benchmark. The training loss and test accuracy curves of the three models are shown in Fig. 6. The parameters of the trapezoidal and normal structure models are shown in Fig. 7.



(a) Training Loss **(b)** Test Accuracy

Figure 6: Training loss and test accuracy

Fig. 6a shows that the three models converge well. Fig. 6b shows that the trapezoidal and normal structure models' accuracies are close on the test set, reaching around 0.89 and higher than the single layer. Fig. 7 displays the parameters of the trapezoidal and normal structure models, showing that the parameters of the trapezoidal structure model are 35.75% less than the normal structure model.

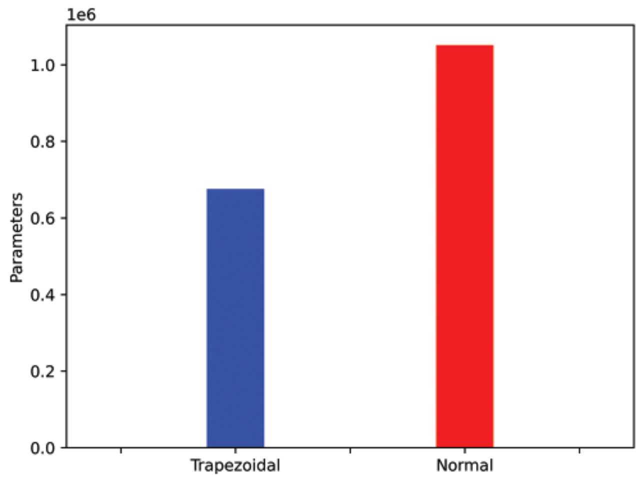
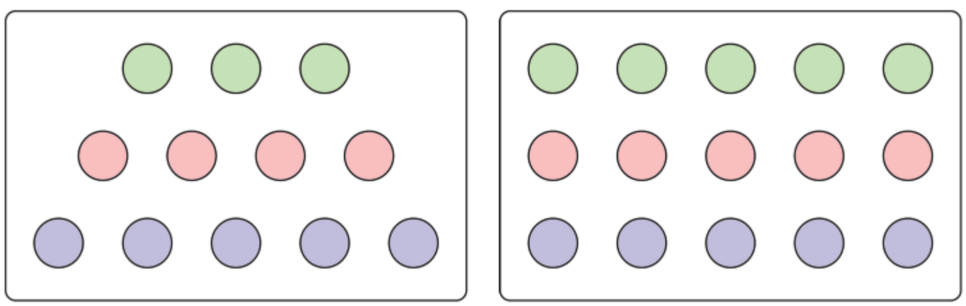


Figure 7: The parameters of trapezoidal and normal structure models

The schematic diagrams of the trapezoidal and normal structure models are shown in Fig. 8. The neurons of the trapezoidal structure model get progressively smaller after the first layer. As a result, the trapezoidal structure model's parameters are smaller than those of the conventional model.



(a) Trapezoidal Structure Model **(b)** Normal Structure Model

Figure 8: The schematic diagrams of the trapezoidal and normal structure models

As illustrated in Figs. 7 and 8, the trapezoidal and normal structure models are comparable. However, the normal structure model has more parameters than the trapezoidal structure model.

4.4 Comparison with Other Works and Discussion

This paper tests the accuracy of the classic models on the SST-2 dataset to verify the advance of the multi-layer Bi-directional LSTM with a trapezoidal structure model. The test results are shown in Table 5.

Table 5: The accuracy of other works

Model	Accuracy
NB	80.7
SVM	78.6
RNN	81.5
LSTM	85.2
Bi-directional LSTM (single layer)	**87.7**
Ours (multi-layer bi-directional LSTM with a Trapezoidal structure)	**89.6**

The multi-layer Bi-directional LSTM with a trapezoidal structure model is more effective for the text sentiment analysis task as it achieved the highest accuracy on the test set. This model performs better on the text sentiment analysis task because of two aspects. First, because this model has a multi-layer structure, it can extract high-dimensional text features, thereby improving the model's accuracy. Then, the number of neurons in each layer is decreased by the sliding window method to reduce parameters. That is, this model achieves higher accuracy with fewer parameters.

The purpose of the research in this paper is to use fewer parameters and obtain higher accuracy; therefore, possible future research directions are:

1. A pre-trained model (such as word2vec [40,41] or bidirectional encoder representations from transformers (BERT) [54]) can be used, combined with the multi-layer Bi-directional LSTM with a trapezoidal structure model; higher accuracy can be obtained by fine-tuning the model [55].
2. This paper sets $N = 128$ and $K = 32$, and future research can test different combinations of N and K to reduce the parameters further [56].

5 Conclusion

This paper starts by outlining the concepts involved in text sentiment analysis. Then, descriptions of the text sentiment analysis algorithms follow. These technologies are based on sentiment lexicon, conventional machine learning, and deep learning, respectively. This is followed by a detailed introduction of the Bi-directional LSTM with a trapezoidal structure model, which includes the general model, embeddings, the trapezoidal structure, the Bi-directional LSTM, training algorithms of this model. The experimental procedure is then completely described, including the data set and data pretreatment, comparison experiments, results, and analysis. The experimental results show that the trapezoidal and normal structure models performed similarly. However, the trapezoidal structure model has 35.75% fewer parameters than the traditional model.

In summary, the trapezoidal structure model achieves higher accuracy with fewer parameters. The strategy proposed in this paper is useful in several fields, such as edge computing and the Internet of Things. Since the proposed model has fewer parameters than the typical structure model, it can operate on these devices more quickly.

Funding Statement: This work is supported by Yunnan Provincial Education Department Science Foundation of China under Grant construction of the seventh batch of key engineering research centers in colleges and universities (Grant Project: Yunnan College and University Edge Computing Network Engineering Research Center).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Birjali, M. Kasri and A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” *Knowledge-Based Systems*, vol. 226, pp. 107134, 2021.

[2] S. M. Mohammad, “Sentiment analysis: Detecting valence, emotions, and other affectual states from text,” in *Emotion Measurement*, Elsevier, pp. 201–237, 2016.

[3] V. S. Agrawal and P. A. Grana, “Evaluation of sentiment analysis of text using rule-based and automatic approach,” *Research & Review: Machine Learning and Cloud Computing*, vol. 1, no. 2, pp. 6–11, 2022.

[4] N. C. Dang, M. N. Moreno-García and F. De la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics*, vol. 9, no. 3, pp. 483, 2020.

[5] E. Fersini, “Sentiment analysis in social networks: A machine learning perspective,” in *Sentiment Analysis in Social Networks*, Elsevier, pp. 91–111, 2017.

[6] M. Wankhade, A. C. S. Rao and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[7] R. Arunachalam and S. Sarkar, “The new eye of government: Citizen sentiment analysis in social media,” in *Proc. of the IJCNLP 2013 Workshop on Natural Language Processing for Social Media (SocialNLP)*, Nagoya, Japan, pp. 23–28, 2013.

[8] S. M. Zavattaro, P. E. French and S. D. Mohanty, “A sentiment analysis of US local government tweets: The connection between tone and citizen involvement,” *Government Information Quarterly*, vol. 32, no. 3, pp. 333–341, 2015.

[9] A. Corallo, L. Fortunato, M. Matera, M. Alessi, A. Camillò *et al.,* “Sentiment analysis for government: An optimized approach,” in *Int. Workshop on Machine Learning and Data Mining in Pattern Recognition*, Hamburg, Germany: Springer, pp. 98–112, 2015.

[10] L. Ziora, “The sentiment analysis as a tool of business analytics in contemporary organizations,” *Studia Ekonomiczne*, vol. 281, pp. 234–241, 2016.

[11] A. H. Alamoodi, B. B. Zaidan, A. A. Zaidan, O. S. Albahri, K. Mohammed *et al.,* “Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review,” *Expert Systems with Applications*, vol. 167, pp. 114155, 2021.

[12] A. K. Jain, S. R. Sahoo and J. Kaubiyal, “Online social networks security and privacy: Comprehensive review and analysis,” *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2157–2177, 2021.

[13] S. Kaisler, F. Armour, J. A. Espinosa and W. Money, “Big data: Issues and challenges moving forward,” in *2013 46th Hawaii Int. Conf. on System Sciences*, Grand Wailea, Maui, Hawaii: IEEE, pp. 995–1004, 2013.

[14] P. Nandwani and R. Verma, “A review on sentiment analysis and emotion detection from text,” *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 1–19, 2021.

[15] N. K. Nguyen, A. -C. Le and H. T. Pham, “Deep bi-directional long short-term memory neural networks for sentiment analysis of social data,” in *Int. Symp. on Integrated Uncertainty in Knowledge Modelling and Decision Making*, Da Nang, Vietnam: Springer, pp. 255–268, 2016.

[16] K. Zhou and F. Long, “Sentiment analysis of text based on CNN and bi-directional LSTM model,” in *2018 24th Int. Conf. on Automation and Computing (ICAC)*, Newcastle University, Newcastle upon Tyne, UK: IEEE, pp. 1–5, 2018.

[17] N. Senthil Kumar and N. Malarvizhi, “Bi-directional LSTM–CNN combined method for sentiment analysis in part of speech tagging (PoS),” *International Journal of Speech Technology*, vol. 23, no. 2, pp. 373–380, 2020.

[18] Y. -Y. Cheng, Y. -M. Chen, W. -C. Yeh and Y. -C. Chang, “Valence and arousal-infused bi-directional lstm for sentiment analysis of government social media management,” *Applied Sciences*, vol. 11, no. 2, pp. 880, 2021.

[19] J. Lin, W. -M. Chen, Y. Lin, C. Gan and S. Han, “Mcunet: Tiny deep learning on IoT devices,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 11711–11722, 2020.

[20] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proc. of the Tenth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Seattle, WA, USA, pp. 168–177, 2004.

[21] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *Proc. of the Fifth Int. Conf. on Language Resources and Evaluation (LREC’06)*, Genoa, Italy, pp. 2200–2204, 2006.

[22] S. Baccianella, A. Esuli and F. Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in *Proc. of the Seventh Int. Conf. on Language Resources and Evaluation (LREC’10)*, Valletta, Malta, pp. 2200–2204, 2010.

[23] P. D. Turney, “Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews,” arXiv preprint cs/0212032, 2002.

[24] S. -M. Kim and E. Hovy, “Automatic identification of pro and con reasons in online reviews,” in *Proc. of the COLING/ACL 2006 Main Conf. Poster Sessions*, Sydney, Australia, pp. 483–490, 2006.

[25] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, “Lexicon-based methods for sentiment analysis,” *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[26] M. Govindarajan, “Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm,” *International Journal of Advanced Computer Research*, vol. 3, no. 4, pp. 139, 2013.

[27] A. Eronen, “Comparison of features for musical instrument recognition,” in *Proc. of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, New Platz, NY, USA: IEEE, pp. 19–22, 2001.

[28] L. Wikarsa and S. N. Thahir, “A text mining application of emotion classifications of twitter’s users using naive bayes method,” in *2015 1st Int. Conf. on Wireless and Telematics (ICWT)*, Manado, Indonesia: IEEE, pp. 1–6, 2015.

[29] L. Dey, S. Chakraborty, A. Biswas, B. Bose and S. Tiwari, “Sentiment analysis of review datasets using naive bayes and k-nn classifier,” arXiv preprint arXiv:1610.09982, 2016.

[30] A. Berger, S. A. Della Pietra and V. J. Della Pietra, “A maximum entropy approach to natural language processing,” *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[31] X. Fei, H. Wang and J. Zhu, “Sentiment word identification using the maximum entropy model,” in *Proc. of the 6th Int. Conf. on Natural Language Processing and Knowledge Engineering (NLPKE-2010)*, Beijing, China: IEEE, pp. 1–4, 2010.

[32] F. Batista and R. Ribeiro, “Sentiment analysis and topic classification based on binary maximum entropy classifiers,” *Procesamiento del Lenguaje Natural*, vol. 50, pp. 77–84, 2013.

[33] A. Sharma and S. Dey, “A boosted SVM based sentiment analysis approach for online opinionated text,” in *Proc. of the 2013 Research in Adaptive and Convergent Systems*, Montreal, Quebec, Canada, pp. 28–34, 2013.

[34] M. Karanasou, C. Doulkeridis and M. Halkidi, “Dsunipi: An SVM-based approach for sentiment analysis of figurative language on twitter,” in *Proc. of the 9th Int. Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado, pp. 709–713, 2015.

[35] K. -X. Han, W. Chien, C. -C. Chiu and Y. -T. Cheng, “Application of support vector machine (SVM) in the sentiment analysis of twitter dataset,” *Applied Sciences*, vol. 10, no. 3, pp. 1125, 2020.

[36] Y. Bengio, R. Ducharme and P. Vincent, “A neural probabilistic language model,” *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[37] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[38] A. Mnih and G. Hinton, “Three new graphical models for statistical language modelling,” in *Proc. of the 24th Int. Conf. on Machine Learning*, Corvalis, Oregon, USA, pp. 641–648, 2007.

[39] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[41] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[42] J. Pennington, R. Socher and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543, 2014.

[43] Y. Kim, "Convolutional neural networks for sentence classification," Eprint Arxiv, 2014.

[44] S. K. Khatri, H. Singhal and P. Johri, "Sentiment analysis to predict bombay stock exchange using artificial neural network," in *Proc. of 3rd Int. Conf. on Reliability, Infocom Technologies and Optimization*, Noida, India: IEEE, pp. 1–5, 2014.

[45] J. Wang, L. -C. Yu, K. R. Lai and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 225–230, 2016.

[46] J. Liu, W. Rong, C. Tian, M. Gao and Z. Xiong, "Attention aware semi-supervised framework for sentiment analysis," in *Int. Conf. on Artificial Neural Networks*, Alghero, Italy: Springer, pp. 208–215, 2017.

[47] A. A. Sharfuddin, M. N. Tihami and M. S. Islam, "A deep recurrent neural network with BiLSTM model for sentiment classification," in *2018 Int. Conf. on Bangla Speech and Language Processing (ICBSLP)*, Sylhet, Bangladesh, 2018.

[48] G. Xu, Y. Meng, X. Qiu, Z. Yu and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *Ieee Access*, vol. 7, pp. 51522–51532, 2019.

[49] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[50] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[51] Z. He, W. Su, Z. Bi, M. Wei, Y. Dong *et al.,* "The improved siamese network in face recognition," in *2019 Int. Conf. on Intelligent Computing, Automation and Systems (ICICAS)*, Chongqing, China: IEEE, pp. 443–446, 2019.

[52] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning *et al.,* "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1631–1642, 2013.

[53] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis *et al.,* "{TensorFlow}: A system for {Large-scale} machine learning," in *12th USENIX Symp. on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, pp. 265–283, 2016.

[54] J. Devlin, M. -W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[55] H. Zhengfang, I. K. D. Machica and B. Zhimin, "Textual similarity based on double siamese text convolutional neural networks and using BERT for pre-training model," in *2022 5th Int. Conf. on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China: IEEE, pp. 107–111, 2022.

[56] O. A. Alzubi, J. A. Alzubi, M. Alazab, A. Alrabea, A. Awajan *et al.,* "Optimized machine learning-based intrusion detection system for fog and edge computing environment," *Electronics*, vol. 11, no. 19, pp. 3007, 2022.