

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/356592652>

Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis

Article in *Applied Sciences* · November 2021

DOI: 10.3390/app112311255

CITATION

1

READS

237

3 authors, including:



Alimarjan Kamyab
Donghua University

7 PUBLICATIONS 35 CITATIONS

[SEE PROFILE](#)



Michael Adjeisah
Donghua University

12 PUBLICATIONS 31 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Twi Corpus: A Massively Twi-to-Handful Language Parallel Bible Corpus for Machine Translation [View project](#)



Sentiment Analysis [View project](#)

Article

Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis

Marjan Kamyab ¹, Guohua Liu ^{1,*} and Michael Adjeisah ²

¹ School of Computer Science and Technology, Donghua University, Shanghai 201620, China; 316027@mail.dhu.edu.cn

² College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China; madjeisah@hotmail.com

* Correspondence: ghliu@dhu.edu.cn

Abstract: Sentiment analysis (SA) detects people's opinions from text engaging natural language processing (NLP) techniques. Recent research has shown that deep learning models, i.e., Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Transformer-based provide promising results for recognizing sentiment. Nonetheless, CNN has the advantage of extracting high-level features by using convolutional and max-pooling layers; it cannot efficiently learn a sequence of correlations. At the same time, Bidirectional RNN uses two RNN directions to improve extracting long-term dependencies. However, it cannot extract local features in parallel, and Transformer-based like Bidirectional Encoder Representations from Transformers (BERT) are the computational resources needed to fine-tune, facing an overfitting problem on small datasets. This paper proposes a novel attention-based model that utilizes CNNs with LSTM (named ACL-SA). First, it applies a preprocessor to enhance the data quality and employ term frequency-inverse document frequency (TF-IDF) feature weighting and pre-trained Glove word embedding approaches to extract meaningful information from textual data. In addition, it utilizes CNN's max-pooling to extract contextual features and reduce feature dimensionality. Moreover, it uses an integrated bidirectional LSTM to capture long-term dependencies. Furthermore, it applies the attention mechanism at the CNN's output layer to emphasize each word's attention level. To avoid overfitting, the Guasiannoise and GuasianDroupout are adopted as regularization. The model's robustness is evaluated on four English standard datasets, i.e., Sentiment140, US-airline, Sentiment140-MV, SA4A with various performance matrices, and compared efficiency with existing baseline models and approaches. The experiment results show that the proposed method significantly outperforms the state-of-the-art models.

Keywords: deep learning; CNN; Bi-LSTM; attention mechanism; social media sentiment analysis; TF-IDF



Citation: Kamyab, M.; Liu, G.; Adjeisah, M. Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis. *Appl. Sci.* **2021**, *11*, 11255. <https://doi.org/10.3390/app112311255>

Academic Editors: Jongweon Kim and Yongseok Lee

Received: 24 October 2021

Accepted: 22 November 2021

Published: 27 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, people express their feelings and opinions to exchange their views using social media, such as Twitter, Facebook, Weibo, LinkedIn, and WeChat. Data gathered from these media has motivated researchers to explore opinion mining and public views. With massive amounts of user-generated text on social media, sentiment analysis (SA) has become an essential part of NLP with many applications, such as information storage and retrieval techniques, web grading, and many more. [1]. Text processing is necessary to remember the ultimate goal of analyzing it and unexpectedly extracting information. Although the amount of data in social media repositories increases exponentially, the traditional algorithms often fail to extract the sentiments from such big data. Researchers recently started to use deep learning (DL) approaches based on the distributed representation to deal with data specifications during training datasets, feature engineering, and other meeting problems on traditional techniques [2]. Studies show that CNN and RNN perform

effectively in DL for several cases, especially in sentiment detection [3]. CNN convolutional applies in NLP to perform the feature mapping and pooling layer applied over the sequence dimension to obtain the fixed-length output. This process enables capturing local features but losing the context information [4]. However, it is also tricky for CNN networks to extract the long-distance features, and the pooling layer cannot capture word location information.

RNN is the state-of-the-art algorithm for sequential data that can solve variable-length sequence input, make the feature representation more valuable, and convert to text vectors, forming a matrix consisting of feature vector and sequence dimensions [5]. However, RNN networks have problems with large-scale parallel computing [6]. Long short-term memory (LSTM) is a variant of RNN with a collection of gates to control the learning flow at each time step to capture long-range dependencies in long sentences [7]. Therefore, LSTM can address long-term information preservation and vanishing gradient text [8]. Compared with the LSTM model, the Bidirectional Long Short Term Memory (BiLSTM) model can significantly analyze a large amount of contextual information from context. The enhanced structure of BiLSTM and bidirectional gated recurrent units (BiGRU) [9] remains the original effect while making the network more straightforward. Furthermore, since Vaswani et al. [10] introduced the Transformer-based method, text classification research has made great strides. To examine this model behavior and provide some human-understandable analysis, the weights of the attention mechanism inherent in these structures have often been considered. Similarly, Song et al. [11] proposed constructing an auxiliary sentence from the aspect and converting ABSA to a sentence-pair classification task that utilized the BERT transformation model. However, the attention RNN models are still generic to deal with both global and local features since the attention mechanism is computed to be determined based on the whole input sequence of the last RNN unit outputs. Meanwhile, fine-tuning strategies for the approaches directly used by the Transformer model remain a critical scientific challenge.

This paper presents a novel attention-based CNN and Bi-LSTM neural network to locate an attention region covering the key sentiment words iteratively to address the problems. First, term frequency-inverse document frequency (TF-IDF) feature weighting and pre-trained GloVe (<https://github.com/stanfordnlp/GloVe> accessed on 20 October 2021), word embedding approaches extracted vital information from textual data. Next, CNN is used to learn high-level feature context from input representation. Simultaneously, the attention mechanism is applied at the CNN output layers to pay suitable attention to different documents. Finally, Bi-LSTM is employed to extract the contextual information from the feature-generated CNN layers to perform the sentiment analysis. To overcome overfitting, we applied Gaussian noise and Gaussian Dropout as regularization in the input layer. We selected four standard English Twitter data because it is an interpersonal interaction site growing with the use of short forms with challenges due to many misspellings, including polysemy and informal language. Social media users commonly post abbreviations and misspellings, so one spelling mistake may change the whole sentence's viewpoint. Likewise, we applied preprocessor tasks that remove noise, lemmatize and estimate special characters. We experimented on a diverse amount of accessible benchmark datasets for our proposed model's effectiveness. Wilcoxon signed-rank test is utilized to verify the existence of significant difference between each pair of methods for sentiment classification. The main contributions are as follows:

- A novel text representation scheme based on TF-IDF feature weighting and pre-trained GloVe word embedding has been presented to extract significant features for sentiment analysis.
- We propose a novel attention-based CNN-Bi-LSTM model to improve accuracy and reduce overfitting. The model adopts the advantages of both CNN and LSTM to improve sentiment knowledge and accuracy. To avoid overfitting, we applied Gaussian noise and Gaussian Dropout.

- The attention mechanism is used to pay suitable attention to different words to improve the feature expression ability.
- We performed a comparative experiment on four Twitter datasets to assess the proposed architecture's effectiveness by improved accuracy.

The rest of the paper is organized as follows: Section 2 discusses the related works. Then, we present the architecture of the framework in Section 3. Next, Section 4 provides the experiment setup and implementation and experimental results and analysis, and finally, Section 5 concludes this work.

2. Related Work

2.1. Traditional Sentiment Analysis

Since sentiment analysis is one of the valuable decision-making methods, most of the work has been done in sentiment classification using data mining, machine learning algorithms, and a knowledge based approach [12]. Rathi et al. [13] insisted that the existing machine learning method has the challenge of providing better sentiment analysis results, so they developed an ensemble method combined from Decision tree and SVM for social media sentiment classification and improved over 2% overall classification performance. Liu et al. [14] employed a two phase supervised learning approach to analyze social media text data. At the first step, lexicon sentiment is performed, and then a product identification model builds to detect the comparative social media content. Finally, they presented the essential advantages of the target product compared to its competitors. These methods are expendable and straightforward. Despite that, the supervised method has severe limitations, such as being dependent on human effort for labeling, long-term activity, and limited effectiveness leading to a conversational and unstructured social media text [15,16].

Several studies show that lexicon-based methods offer more portable solutions across the domains. Still, these approaches are usually less accurate [17,18]. Another major challenge of sentiment analysis on machine learning and lexicon-based, including the hybrid method, is feature selection, typically domain-dependent. Song et al. [19] proposed a text representation model named Word2PLTS that used linguistic terms sets (PLSTSs) short text sentiment analysis by combined supervised machine learning and unsupervised machine learning. The Word2PLTS model achieves a positive impact in solving the problems of data unavailability and data sparsity. However, the model is very complex and can only be applied to a short text. DL is known for the multiple representation learning levels in machine learning and has recently been applied to sentiment analysis with significant results [20].

2.2. Weighted Word Embedding for Sentiment Analysis

Word Embedding is converting text into numbers that can be readable by machine learning and DL algorithms such as Co-Occurrence Vector, Bag of words [21], TF-IDF Vector [22], LDA [23]. Continuous vector representations of words algorithms such as Glove and Word2Vec deep learning techniques can convert words into meaningful vectors. Fu et al. [24] used Word2Vec for word representations of English and Chinese Wikipedia datasets. The word representation is applied as inputs of the recursive autoencoder for the sentiment analysis approach. Qin et al. [25] deal with a relation classification task utilizing the CNN approach to automatically control feature learning from raw text. They used pre-trained Word2Vec as inputs of CNN for data-driven tasks. Abid et al. [26] accomplished empowering subjectivity knowledge in sentiment analysis by merging the Glove with RNN and CNN for weight pooling. They employed CNN to capture global features with conventional pooling methods successively. The proposed architecture attained 87.18% best accuracy on STSC-1.5k datasets by the Glove-Bi-GRU-CNN model. Jianqiang et al. [27] introduced the GloVe-DCNN method which obtained unsupervised learning based on a large Twitter dataset for training and predicting sentiment classification labels. This method combined word sentiment polarity score and n-grams features to form

a sentiment feature set of tweets and the features fed into the CNN layer. The authors evaluated the results on five Twitter datasets which achieved significant results compared to the state-of-the-art. Notably, most of the above techniques may lead to a lack of semantic information, problems of high dimensionality, high sparsity, and ignoring text sentiment information [28]. In addition, increasing the accuracy of pre-trained word embedding is essential and plays a vital role in sentiment classification. Studies show that combining pre-trained word embedding such as Word2Vec and GloVe vectors in their DL model decreases accuracy [29]. In addition to the above methods and difficulties, Zhao and Mao [30] proposed a bag of word method with the help of term frequency (TF) weighting algorithms and Word2vec, which achieve significant results for word representation. Feature weighting and word vector combination also perform meaningful words' extraction tasks [31,32].

2.3. Deep Models for Sentiment Analysis

DL models used for different categories of applications have become the standard tool for solving computer vision problems [33]. Dang et al. [34] analyze 32 of the latest research articles that have employed DL to address the sentiment classification issues. RNN, DNN, and CNN architectures were analyzed and combined with the used TF-IDF and word embedding to transform input data to a DL model then predict the sentiment analysis. In this study, several experiments were conducted on a different dataset, including the Twitter dataset, to evaluate the DNN, CNN, and RNN models' performance. authors conclude that deep learning techniques combined with word embedding are more reliable than combined with TF-IDF.

Johnson et al. [35] proposed a word-level CNN model and found that deepening convolutional can improve the modeling effect. Rezaeinia et al. [36] introduced an improved word vector (IWV) method based on POS, lexicon approach, glove pre-trained word embeddings, and employed CNNs. The model experiment evaluation shows that the accuracy of pre-trained word embedding improved significantly compared to using the baselines. However, these models, including [27,37], faced vanishing gradient problems and did not consider long dependencies.

Due to long-term dependencies and vanishing gradient problems, the researcher uses RNN and its variants extensively in sentiment analysis. Wang et al. [38] proposed a regional CNN (CRNN) model. First, the regional CNN applied to the input vectors considers a whole text as input. Then, max pooling is used to decrease the dimensionality of the local feature. Finally, the results are sent into the LSTM layer to extract the long dependencies across sentences. Kim and Yoon [39] proposed a novel SA framework by a new correlation of CNN and BiLSTM in a statistical manner. They created a multi-domain word embedding dictionary by employing sentiment lexicon and word2vec with a CNN layer to perceive the basic features and feeds the output into the BiLSTM. Experimental results showed 1–4% performance improvement.

Nguyen and Nguyen [40] presented a CNN and LSTM to capture local dependencies and memorize long-distance information. The authors integrated the advantage of various deep models to reduce overfitting in training. Chatterjee et al. [41] proposed a deep model named Sentiment and Semantic-Based Emotion Detector (SS-BED). The model uses two LSTM layers and two different word embeddings matrices to extract sentiment and semantics for emotion recognition. However, besides the strengths and weaknesses mentioned for those models, there is a general drawback for all that cannot consider each sentence's importance differently.

With the advent of powerful attention mechanisms based on weighted transformation, sentiment classification has significantly improved on mentioned problems. For example, Attention-based BiLSTM with convolution layer (AC-BiLSTM) [42] offers the model to deal with high dimensionality and sparsity of text data challenges. The 1D-CNN layer extracted the n-gram features of input data to reduce text data dimensionality. BiLSTM is applied at the output of the CNN layers. Even though their work results are encouraging, the research did not fully address the co-occurrence of short and long dependencies. Wen and Li [43]

proposed a combination of RNN and CNN with attention mechanism called ARC to classify tweets and reviews. This model is designed to extract local n-gram and global features on sequential information by feeding the output of a one-layer bidirectional GRU into CNN. Basiri et al. [44] introduced an attention-based bidirectional CNN–RNN deep model (ABCDM) using two independent BiGRU and BiLSTM layers. It extracts both past and future contexts by considering temporal learning flow in both directions. Jing et al. [45] proposed bidirectional LSTM (SAMF-BiLSTM) into account to design a self-attention system and multi-channel features. The model effectively delivered a relationship of sentiment polarity of each word in the sentence with target words. The classification efficiencies by this technique were higher than the traditional methods in various aspects. However, the authors stated that this model needs to redesign the mechanism for a particular document-level classification and did not consider overfitting problems.

Latterly, Kumawat et al. [46] introduced a transformer-based deep learning model to correct context interpretation due to the lack of labeled social network datasets. These models were evaluated and performed on the Twitter US-Airline Sentiment dataset and achieved 0.812% accuracy. Likewise, Sun et al. [47] proposed constructing an auxiliary sentence from the aspect and converting ABSA to sentence–pair classification tasks that utilized the BERT transformation model. However, these methods simply employed the BERT model as a black box in an embedding layer for encoding the input sentence. Additionally, the method also has the drawback of the extraction of contextual information. Recently, Wang et al. [48] proposed the Entailment as Few-Shot Learner (EFL) approach to turn small LMs into better few-shot learners and convert the class label into a natural language sentence. The method established 1.9 pt absolute improvement compared to standard fine-tuning of the RoBERTa model and an average of 19 pt.

improvement compared to the standard fine-tuning method. Nevertheless, the model fails to consider the importance of each label based on reinforcement learning. The following section presents our proposed model to solve the word embedding and DL mentioned problems.

3. Proposed Architecture

This section introduces the structure of our proposed model ACL-SA. The overall architecture of the ACL-SA is shown in Figure 1. It comprises data processing, Weighted word representation, CNN layers, attention layer, bidirectional LSTM, Fully connected, and output layer.

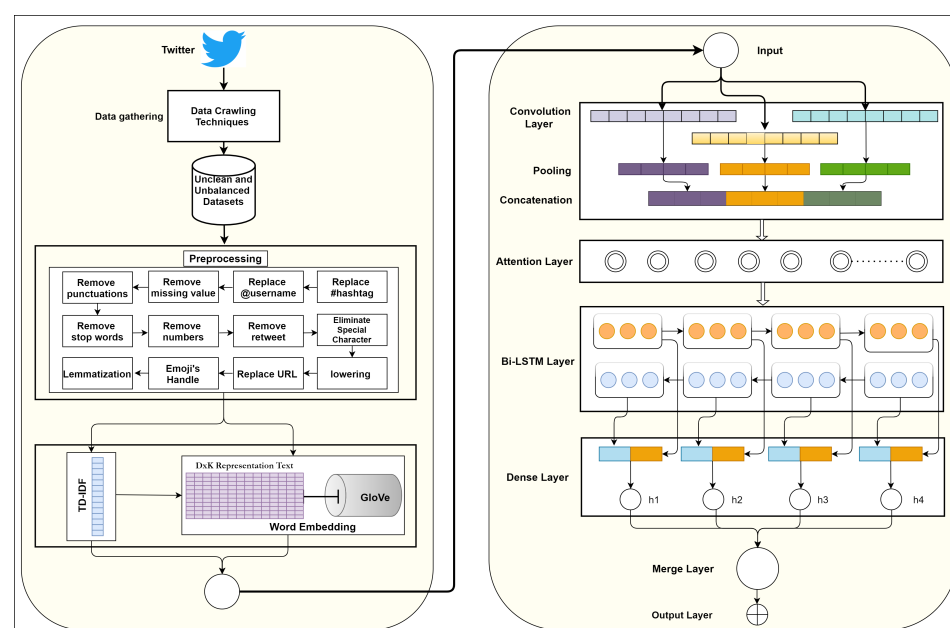


Figure 1. Model architecture.

3.1. Data Preprocessor

Preprocessing is a crucial step in making the text more digestible by removing non-sense phrases, noise, and unnecessary repetitions so that a deep model can boost their efficiency. Perversely, the language used on social media is nonstandard and informal, and noise requires extensive processing before feeding the network. Therefore, we employ various preprocessing techniques such as lemmatization, removing non-Unicode, non-English characters, replacing URLs, and User Handlers to handle all noise, and make text ready for training.

3.2. Weighted Word Representation

Term frequency-inverse document frequency (TF-IDF): Studies show that the weighted averages of word embedding can improve unsupervised NLP tasks, especially sentiment classification performance [49]. TF-IDF is an unsupervised term weighting scheme for information retrieval and text mining. TF-IDF represents the relative frequency of a word t in a text document, and inverse document frequency scales with the number of documents. It can process as follows:

$$W_{d,t} = tf \times \log\left(\frac{N}{df}\right) \quad (1)$$

W indicates the weight value for term t , and document d , the total number of the document in the corpus defines as N , tf , and df denote the term frequency indicating the number of times and number of the document in a particular term, respectively.

The global vectors (GloVe): GloVe is a word2vec-based word representation to learn word embeddings from text documents effectively [50]. A pre-trained word embedding model GloVe with 2 billion tweets, 27 billion tokens, and 1.2 million vocabularies was used to generate a word vector matrix as a 200-dimensional vector. TF-IDF with GloVe models utilized to observe the proposed model performance. If each input k -word is represented as $T(t_1, t_2, \dots, t_k)$, then each word is converted into word vector of d dimension. Hence, R^d will be the dimensions space of each word; then, each input text is represented as $R^{k \times d}$ dimension space and input text matrix generation as $T = \{t_1, t_2, \dots, t_k\} \in R^{k \times d}$. Finally, the feature vector f_v for the T document concatenated \oplus with word embedding is shown as follows:

$$f_v = w_1 \oplus w_2 \oplus w_3 \dots \oplus w_{n-1} \oplus w_n \quad (2)$$

To improve the text representation, we integrate pre-trained Glove word embedding with TF-IDF weighing.

$$V_i = W_{d,t} \times f_v \quad (3)$$

where f_v , as mentioned above, is a word vector matrix obtained by Glove, and $W_{d,t}$ is the TF-IDF weighing of document d and term t . This method can solve the dimensional problem of the high dimensional sparse matrix. We also applied Gaussian Noise and Gaussian Dropout after receiving text representation from the word representation layer. The Gaussian Noise and Gaussian Dropout process can be used as a regularization method, making the model more vital and less prone to overfitting. Similarly, since Gaussian noise is applied directly to the word embedding, this process serves as a random data augmentation during training time.

3.3. Attention Based Deep Layers

To utilize the knowledge from the weighted word representation layer, we used three region CNN networks, given the word representation as input, and fed into a convolutional layer based on each tweet. The convolutional word vector matrix is calculated through an F_n weighted matrix defined as $w \in R^{t \times m}$ frequently to capture the inherent and local features, and t is a word vector selected in the matrix of F_n as follows:

$$h_i = f(V_{i:j+t-1} \times W[i] + b_i) \quad (4)$$

where f is the activation function for the nonlinearity, W represents the matrix's weight, b is the bias, and $h \in R_{n-t+1}$ is for the generation of a feature map by a t word vector. Once convolutional layers produce feature maps, the max-pooling layer applies to minimize the dimensions of the dataset and extracts the most important features as shown in the equation:

$$p_i = \text{Max}[h_i] \quad (5)$$

where the $P_i \in R^{n-t+1/2}$ is the feature map obtained after the max-pooling layers. Max-pooling is applied to features from the different CNN layers filter to extract equally essential features, but it is not focused on semantics and polarity importance. Therefore, we devote attention to emphasize the importance of each feature on the CNN-generated features as follows:

$$A_i = \frac{\exp(p_i)}{\sum_i \exp(p_i)} \quad (6)$$

The above equation shows the calculation of max-pooling attention, and A_i is the generated attention score for each feature context p_i [12].

The Bi-LSTM network is applied to the attention score's output to learn the feature context. We use Bi-LSTM to generate final features by sequentially processing the map. The last feature map is obtained from the final feature context p_i from CNN and A_i attention scores.

We used Bidirectional LSTM to consider both forward and backward features parallel and simultaneously concatenate the hidden state of two LSTMs representing each position. Equation (7) represents forward, and (8) represents backward LSTMs:

$$\vec{h}_{t_{lstm}} = \overrightarrow{LSTM}(c_{t-1}, h_{t-1}, A_i) \quad (7)$$

$$\overleftarrow{h}_{t_{lstm}} = \overleftarrow{LSTM}(c_{t-1}, h_{t-1}, A_i) \quad (8)$$

where c_t and h_t represent the hidden states and memory cell, respectively; however, the h_{t-1} and c_{t-1} represent the previous states of LSTM function, and A_i is the attention score as an input vector in the LSTM network. We now obtain an annotation for each input vector by concatenating the forward and backward context in Equation (9):

$$h_{t_{lstm}} = LSTM[\vec{h}_{t_{lstm}}, \overleftarrow{h}_{t_{lstm}}] \quad (9)$$

$h_{t_{lstm}}$ is the concatenating output of forwarding ($\vec{h}_{t_{lstm}}$) and backward ($\overleftarrow{h}_{t_{lstm}}$) extracts a long dependencies feature. The extracted feature of the entire sentence is $[\vec{h}_{t_{lstm}}, \overleftarrow{h}_{t_{lstm}}]$. In this way, the forward and backward contexts can be considered simultaneously.

3.4. Full Connection and Output Layer

The fully connected dense layer is used to transform the bidirectional network into high-level sentiment representation to predict text sentiment polarity. The output is obtained as follows:

$$h_i = \text{Relu}(w_i h_p + b_i) \quad (10)$$

where w_i , and b_i are parameters that are learned in training, h_i is obtained features, and h_p is the feature map received from the Bi-LSTM network. The output layer performs the sentiment classification using the merge feature layer, shown in Figure 1. Here, sigmoid and Sigmoid classifiers are used for binary and multiclass datasets, respectively. Cross entropy is used to compute the discrepancy between the predicted and actual sentiment of the text.

4. Experiments and Analysis

This section introduces our experiments and explains the results in detail. We test our model on four English Twitter datasets, considering the ultimate goal of accurately

analyzing our proposed methods. Finally, the developed models are compared with the existing research to examine the proposed model's predictive performance.

4.1. Datasets

The Twitter datasets for our empirical analysis include Sentiment140 (<http://help.sentiment140.com/for-students> accessed on 20 November 2021), US-Airline (<https://www.kaggle.com/crowdflower/twitter-airline-sentiment> accessed on 20 November 2021), Sentiment140-MV, Sentiment Dataset for Afghanistan (SD4A) (<https://www.kaggle.com/kamyab20/sentiment-dataset-for-afghanistan-sd4a> accessed on 20 November 2021). The above datasets have been widely used in sentiment analysis tasks so that the experiment results have an accurate evaluation. The Sentiment140 dataset is scraped from Twitter by Sandford graduate students [51]. Each tweet is labeled positive and negative according to its emotional sentences, and there are 248,576 positives and 80,000 negatives. Currently, this dataset is one of the most used standard datasets for text classification [19,34,36–38,40,44]. The US-Airline dataset is available on the Kaggle website, which contains 14,641 tweets. This dataset was collected about the US airline problem in 2015 and divided into positive, negative, and neutral sentiment categories [27,34,41–43]. After removing neutral tweets, we obtain 11,541 tweets for this research. Sentiment140-MV is the modification version (MV) of sentiment140 with 18,309 tweets. Abid and Alam [26] modified the sentiment140 dataset into 4000 tweets and 1500 tweets individually. We pick more tweets to see the effect of our model. SD4A is collected by Kamyab et al. [52] about Afghanistan's security status from 29 March 2018 to 21 June 2018. It contains 18,309 positive and 18,539 negative tweets. Detailed statistics of the dataset are listed in Table 1.

Table 1. Detailed statistics of the dataset: We used four different datasets to verify our model's effectiveness.

Dataset	Max	Min	Avg	Positive	Negative	Total
SD4A	38	4	16	18,309	18,539	36,848
Sentiment140	40	1	12	248,576	80,000	1,048,576
Sentiment140-MV	35	3	15	11,628	6585	18,213
US Airline	30	1	15.5	2343	9112	11,455

4.2. Experimental Setup

The inputs to the proposed model are embedding initialization by combining TF-IDF weighting and Glove with 200 dimensions with other parameters during the network training steps. At the input layer, we used Gaussian noise with the value of 0.5 and Gaussian dropout with the value of 0.3 at the connection network with the one-dimensional convolutional layer to avoid overfitting. For CNN's layers, we apply three channels with 600, 300, and 150 filters with kernel window sizes of k (3, 4, and 5) and Relu as an activation function to each convolutional layer. The output of the convolutional layer then fits into max-pooling with pool size 2. After receiving the feature from the concatenation layer, we applied the Attention mechanism before feeding the output to the BiLSTM layer. The BiLSTM consists of 256 batch size, a dense size of 128, Relu, and kernel regularization rate of 0.0001. After the dense layer, a sigmoid function is used for the binary classifier. Finally, we engaged binary cross-entropy to train the model and Adam optimizer for the model's learning rate. We run the experiment on Windows 10 on an Intel core i5 processor with 3.00 GHz and 16 GB of RAM.

4.3. Model Variation and Baselines Method

We used several recent similar models for each dataset developed for sentiment classification. We pick the same results mentioned in the baseline reference paper. Moreover, five model variations are tested during the experiment: CNN and LSTM, join models of CNN and LSTM with the TF-IDF weighting, and trained model glove word representation, defined as Table 2.

Table 2. Proposed method using a pre-trained word vector GloVe.

Proposed	Narrative
TF-IDF-Glove-CNN	Methods used TF-IDF-Glove word embedding.
TF-IDF-Glove-LSTM	
TF-IDF-Glove-BiLSTM	
TF-IDF-Glove-CNN-LSTM	
TF-IDF-Glove-CNN-BiLSTM	

4.4. Results Analysis and Discussion

This section provides baseline comparisons to evaluate the model accuracy, loss value, and effectiveness in minimizing overfitting. Moreover, we applied five different variation models using TF-IDF-glove word representation and various deep learning algorithms with a batch size of 256, epochs remaining 50, and the learning rates to 0.001 for all the datasets for satisfactory accuracy.

4.4.1. Analysis of Results on the Sentiment140 Dataset

Table 3 demonstrates our proposed model's comparisons with different models offered by various authors. An overall glance for the sentiment140 Twitter data shows that these given models acquired 56.95% to 87.12% accuracies. This table depicts our proposed model's highest accuracy with 87.12%, while TF-IDF-RNN [34] achieved the lowest accuracy.

Table 3. Sentiment140 dataset and the baseline models' accuracy comparison.

Author/Year	Models	Accuracy
Li et al. [53]	RNTN	0.8070
Li et al. [37]	CharSCNN	0.8570
Wang et al. [38]	CRNN	0.7987
Nguyen & Nguyen [40]	CNN autoencoder	0.7931
Nguyen & Nguyen [40]	BiLSTM autoencoder	0.7911
Rezaeinia et al. [36]	IWV	0.8052
Song et al. [19]	SAPCP	0.8522
Dang et al. [34]	TF-IDF-CNN	0.7668
Dang et al. [34]	TF-IDF-RNN	0.5695
Dang et al. [34]	Word embedding -CNN	0.8006
Dang et al. [34]	Word embedding -RNN	0.8281
Basiri et al. [44]	ABCDM	0.8182
Subba and Kumari [54]	BERT-GloVe-Word2Vec-BiRNN	0.84
Wang et al. [48]	EFL	0.863
Our model	TF-IDF-Glove-LSTM	0.8234
	TF-IDF-Glove-BiLSTM	0.8345
	TF-IDF-Glove-CNN	0.8322
	TF-IDF-GloveLSTM+CNN	0.8432
	TF-IDF-Glove-BiLSTM-CNN	0.8530
	ACL-SA	0.8712

4.4.2. Analysis of Results on the US-Airline Dataset

Table 4 concludes the accuracy of different model performances on the US Airline dataset compared to our proposed model. We found that our proposed model has a significant accuracy of 94.01% for this dataset, with RNN-TF-IDF [34] remaining the lowest compared to all other models. Similarly, we found that TF-IDF-Glove's performance with different neural networks for the US Airline dataset is noteworthy. For example, TF-IDF-Glove-BiLSTM-CNN attained 93.5% accuracy.

Table 4. US-Airline Twitter dataset accuracy compression with baseline models.

Author/Year	Models	Accuracy
Li et al. [53]	CharSCNN	0.865
Kumawat et al. [46]	BERT-DNN	0.81
Wang et al. [38]	CRNN	0.9205
Yang et al. [55]	HAN	0.9035
Jianqiang et al. [27]	Glove-DCNN	0.839
Wen & Li [43]	ARC	0.9229
Rezaeinia et al. [36]	IVM	0.8985
Chatterjee et al. [41]	SS-BED	0.91
Liu & Guo [42]	AC-BiLSTM	0.9172
Dang et al. [34]	TF-IDF-CNN	0.6879
Dang et al. [34]	TF-IDF-RNN	0.6174
Dang et al. [34]	Word embedding-CNN	0.8236
Dang et al. [34]	Word embedding-RNN	0.8376
Basiri et al. [44]	ABCDM	0.9275
Wang et al. [48]	EFL	0.9208
Our model	TF-IDF-Glove-LSTM	0.9041
	TF-IDF-Glove-BiLSTM	0.913
	TF-IDF-Glove-CNN	0.9198
	TF-IDF-Glove-LSTM+CNN	0.9189
	TF-IDF-Glove-BiLSTM-CNN	0.9358
	ACL-SA	0.9401

4.4.3. Analysis of Results on the Sentiment140-MV Dataset

Table 5 illustrates the comparative analysis of the proposed model's performance using two models. One uses the TF-IDF feature method with four different machine learning classifiers, and the second employs the TF-IDF-Glove method with five model variation neural networks. The Sentiment140-MV dataset is a modified form of the Sentiment140 dataset with 18,000 tweets. We applied these previous machine learning and deep learning methods on this dataset to compare our model's performance with others. In comparison, our proposed model outperformed the other models. Meanwhile, deep learning models acquired 91.94% average accuracy; however, the machine learning model achieved 81.25% average accuracy with the Sentiment140-MV dataset. It enables us to conclude that our proposed model attains adequate accuracy with this modified dataset.

Table 5. Sentiment140-MV accuracy compression with baseline models and our proposed models.

Models	Accuracy	Average Accuracy
TFIDF with DT	0.7568	0.812475
TFIDF with RF	0.8589	
TFIDF with SVM	0.8621	
TFIDF with NB	0.7721	
TF-IDF-Glove-LSTM	0.9075	0.9194
TF-IDF-Glove-BiLSTM	0.9090	
TF-IDF-Glove-CNN	0.9141	
TF-IDF-Glove-LSTM-CNN	0.9369	
TF-IDF-Glove-BiLSTM+CNN	0.9370	
ACL-SA	0.9443	0.9443

4.4.4. Analysis of Results on the SD4A Dataset

Table 6 presents the same comparison of models as we have evaluated in Table 5; however, in this evaluation, we utilized our dataset SD4A, which has never been used previously. The DL models almost gained 92.07% average accuracy on our newly SD4A dataset, similar to the Sentiment14-18k dataset. However, the machine learning classifier with the TF-IDF model secured 82.82% average accuracy with the SD4A dataset, 1.58% higher than

Sentiment140-MV. In contrast, our proposed model's accuracy for our constructed data is 94.53%, sufficiently higher than the existing neural networks.

Table 6. SD4A accuracy compression with baseline models and our proposed models.

Models	Accuracy	Average Accuracy
TFIDF with DT	0.7391	0.828225
TFIDF with RF	0.8592	
TFIDF with SVM	0.8621	
TFIDF with NB	0.8525	
TF-IDF Glove-LSTM	0.9141	0.9207
TF-IDF Glove-BiLSTM	0.911	
TF-IDF Glove-CNN	0.9209	
TF-IDF Glove-LSTM-CNN	0.9238	
TF-IDF Glove-BiLSTM+CNN	0.9337	
ACL-SA	0.9453	0.9453

For simplicity, Table 7 and Figure 2 summarize the proposed model's accuracy with the aforementioned neural networks and datasets. It depicts that the proposed ACL-SA model successfully attained the highest accuracy with our structured dataset. In comparison, the average accuracy comparison of our proposed model with the existing models for all datasets is differentiated with the following inequality expression:

Proposed Model (92.52) > TF-IDF-Glove-BiLSTM+CNN (91.48) > TF-IDF-Glove-LSTM+CNN (90.82) > TF-IDF Glove-CNN (89.67) > TF-IDF Glove-BiLSTM (89.18) > TF-IDF Glove-LSTM (88.72).

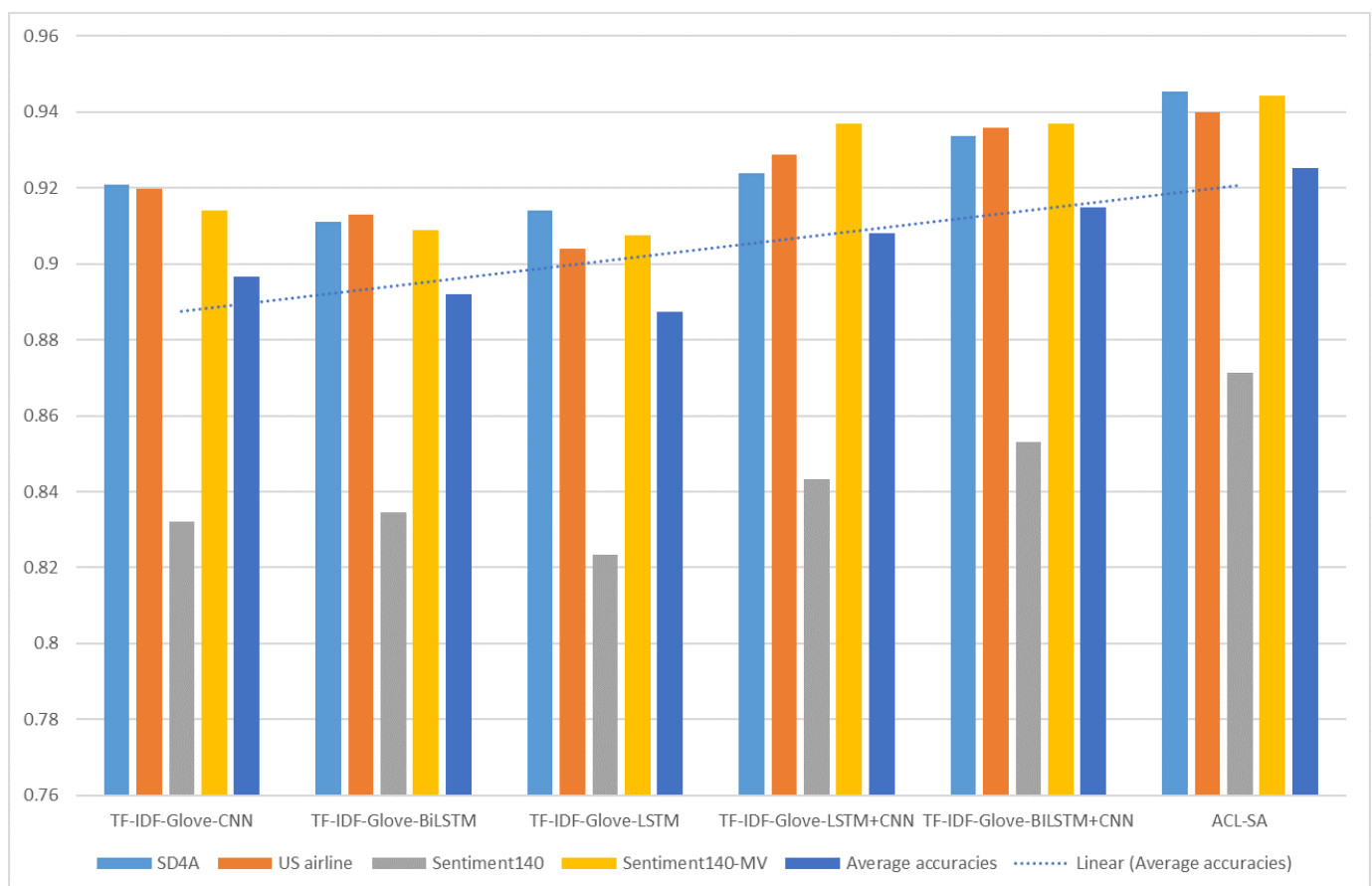


Figure 2. Comparative analysis of the proposed model and model variation on different datasets.

Table 7. Summary of the proposed model's accuracy.

Dataset	TF-IDF-Glove-CNN	TF-IDF-Glove-BiLSTM	TF-IDF-Glove-LSTM	TF-IDF-Glove-LSTM+CNN	TF-IDF-Glove-BiLSTM+CNN	ACL-SA
SD4A	0.9209	0.911	0.9141	0.9238	0.9337	0.9453
US airline	0.9198	0.913	0.9041	0.9289	0.9358	0.94
Sentiment140	0.8322	0.8345	0.8234	0.8432	0.853	0.8712
Sentiment140-MV	0.9141	0.909	0.9075	0.9369	0.937	0.9443
Average accuracies	0.8967	0.8919	0.8873	0.9082	0.91499	0.9252

Furthermore, to verify the significance of the experimental results obtained by this study, the non-parametric tests are used with a confidence interval of 95% in terms of classification accuracy. Significant differences are found in each pair for the test results obtained of the ACL-SA with CRNN, IWC, ABCDM, and Word Embedding-RNN for the Sentiment140 dataset, Table 3. Similarly, the Wilcoxon Signed-rank test was used in each pair ACL-SA with ABCDM, CRNN, ACR, IVM, SS-BED, and Word Embedding-RNN of the US-airline dataset, Table 4. The results are shown in Table 8. In both Tables 3 and 4, the Wilcoxon test results show an enormously significant difference, and the hypothesis has been validated (p -value < 0.05) for all methods paired with our proposed method.

Table 8. Wilcoxon tests of different sentiment analysis methods in terms of classification accuracies.

Measure	Dataset	Comparison	Hypothesis	p -Value
Classification accuracy	US-airline	ACL-SA vs. ABCDM	Rejected for ACL-SA	0.003
		ACL-SA vs. CRNN	Rejected for ACL-SA	0.025
		ACL-SA vs. ACR	Rejected for ACL-SA	0.025
		ACL-SA vs. IVM	Rejected for ACL-SA	0.026
		ACL-SA vs. SS-BED	Rejected for ACL-SA	0.024
		ACL-SA vs. AC-BiLSTM	Rejected for ACL-SA	0.016
		ACL-SA vs. Word Embedding-RNN	Rejected for ACL-SA	0.031
	Sentiment140	ACL-SA vs. CRNN	Rejected for ACL-SA	0.001
		ACL-SA vs. IWV	Rejected for ACL-SA	0.011
		ACL-SA vs. SAPCP	Rejected for ACL-SA	0.001
		ACL-SA vs. Word Embedding-RNN	Rejected for ACL-SA	0.001
		ACL-SA vs. ABCDM	Rejected for ACL-SA	0.005

As mentioned in the literature [40,45], overfitting, a critical challenge in deep learning models, reduces the model's accuracy and performance. However, this challenging task is resolved by utilizing the Gaussian Noise and Gaussian Dropout on different Twitter datasets. Figure 2 presents the convergence of our proposed model that attained a reliable accuracy based on experiments conducted on training and validation datasets with 50 epochs and a batch size of 256. For instance, in Figure 3a, the SD4A validation data are initiated from an accuracy of 0.814, steadily increasing to 0.935 at epoch 10, and remains constant for all other epochs.

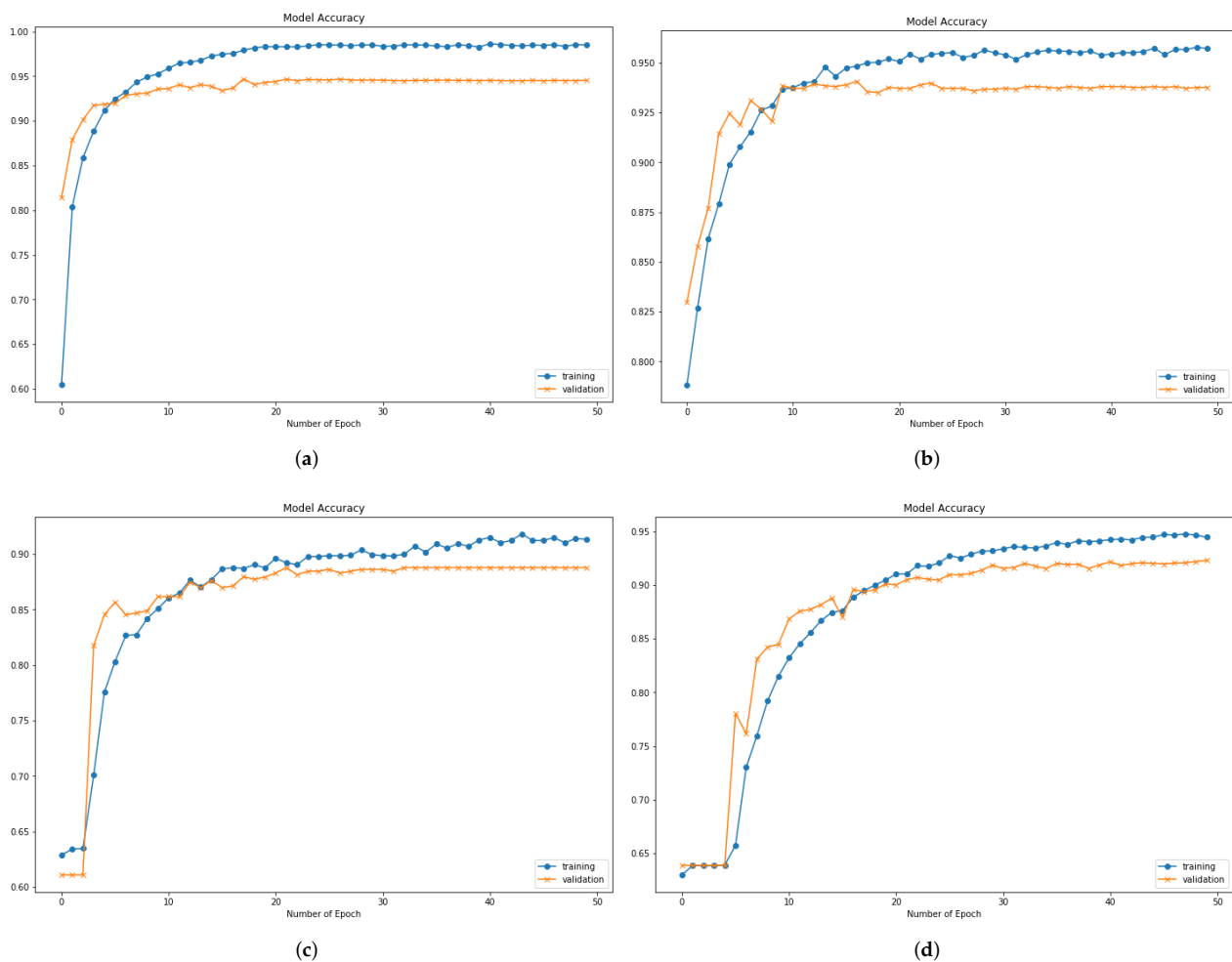


Figure 3. Performance comparison obtained by different methods on the SD4A dataset (a); US-Airline dataset (b); Sentiment140 dataset (c); Sentiment140-MV dataset (d).

Meanwhile, the training data accuracy fluctuates from 0.60 and increases steadily to 0.952 with epoch increment. At the same time, it became constant after the 15th epoch. Similarly, Figure 3b presents the accuracy trend of training and validation of the US airline dataset. It provides a rapid accuracy curve of 10 epochs starting from 0.839 to 0.938 on the validation set. For the training set, it starts from 0.788 to 0.94, which becomes constant after 10 epochs. Figure 3c,d show that the accuracy trends for the sentiment140 and sentiment140-118 k datasets are slightly different from SD4A and US-airline datasets. Figure 3c shows that the accuracy trends for the sentiment140 dataset improved smoothly from 0.75 to 0.89 for epochs 5 to 20. Later, it is constant on average for both training data and validation data. On the other hand, in Figure 3d, we observed that the training and validation results of the Sentiment140-MV dataset could not attain accuracy in epochs 0–5. Later, it increased with the number of epochs and became constant after epoch 30. Apart from these analyses, our proposed model significantly combatted overfitting in all different datasets. As a result, the training dataset's accuracy is slightly higher after the 20th epoch than the validation or test dataset in all datasets—while, for initial epochs, the test data's accuracy is a bit higher. Figure 4 likewise shows the loss value to confirm the combatted overfitting effect of our model on the four Twitter datasets. From the figure, the loss of our models in training and validation sets shows a decreasing trend with the increase of epochs and finally reaches a lower value and becomes stable. For instance, in Figure 4a, the loss of SD4A validation data starts from 0.5106, steadily decreasing to 0.1764 for 10 Epochs, and then becomes stable for all other iterations. Figure 4b presents the loss trend of training and validation

of the US-airline dataset. It provides a rapid loss curve of 10 iterations starting from 0.4669 to 0.1835 on the validation set and becomes stable. Figure 4c,d belongs to the loss rate of sentiment140 and Sentiment140-MV, respectively. The loss validation rate starts from 0.6919, decreasing the trend to 0.3293 for sentiment140. Similarly, loss validation for Sentiment140-MV starts from 0.7091 and steadily decreases to 0.2289. In both, the loss trends for the validation set become stable after 20 epochs.

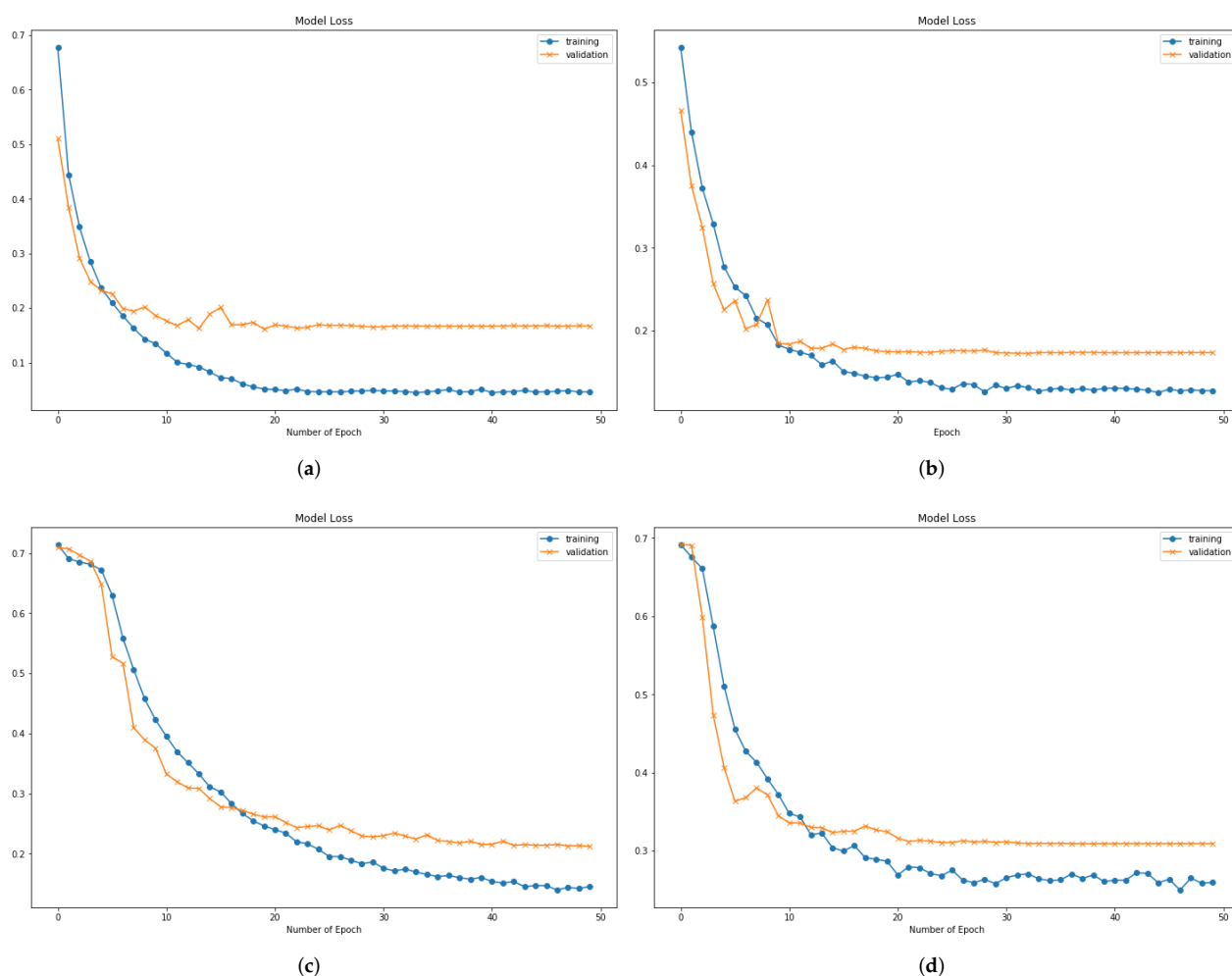


Figure 4. SD4A loss rate change on training and validation data (a); US-Airline loss rate change on training and validation data (b); Sentiment140 loss rate change on training and validation data (c); Sentiment140-MV loss rate change on training and validation data (d).

In the result evaluation of Figures 3 and 4, our proposed ACL-SA network model converges after the mentioned number of epochs with consistent accuracy and loss. Thus, these analyses provide evidence that the ACL-SA network reduces the overfitting problem and attains adequate accuracy. Our proposed model performs better than the other baseline algorithms on all datasets based on the experimental and comparative analyses. Furthermore, the model minimizes the losses and increases the accuracy of the model.

5. Conclusions

This work presents a novel ACL-SA model to tackle the lack of semantic information, high dimensionality, and overfitting problems. Our model is a joined CNN- bidirectional-LSTM architecture that uses a combination of TF-IDF weighting and pre-trained Glove word embeddings. In addition, we engaged three CNN layers to extract contextual features with Max-pooling at the output of the CNNs layer for dimensionality reduction of feature

space. Furthermore, an attention mechanism is employed at the end of the CNN layers to put more or less attention into different words. Collectively, the bidirectional LSTM network serves as a temporal feature and updates the CNN output layer on the past and future sentiment representation. Finally, we employed a dense and fully connected layer with Relu activation and sigmoid function to transform the vector into sentiment polarity classification. In addition, we present TF-IDF weighting and pre-trained word embedding to extract significant word representation to keep the word order information of the Twitter data. Gaussian Noise and Gaussian Dropout were subsequently used on the input layer to overcome overfitting. Experiments were conducted on four different Twitter datasets to analyze the performance of the proposed model. We engaged the most recent deep learning research models for sentiment analysis for comparison. The proposed model's performance achieved a magnitude of 0.9453, 0.993, 0.94, and 0.8712 on the SD4A, Sentiment140-MV, US-Airline, and Sentiment140 datasets—a significantly improved accuracy compared to baseline methods results. Wilcoxon test results verified the significant difference between the proposed model and baselines. The hypothesis has been validated (p -value < 0.05) for all methods pair tests with our proposed method. In the future, we will work to expand our model for other languages.

Author Contributions: Conceptualization, resources, M.K. and G.L.; methodology, implementation, validation, investigation, data curation, visualization, M.K.; formal analysis, paper draft preparation, writing and editing, M.K. and M.A.; supervision, funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Innovation and Development of Shanghai Industrial Internet (Grant No. XX-GYHL-01-19-2527).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets presented in this study can be found in the following links: Sentiment140- <http://help.sentiment140.com/for-students>; US-Airline-<https://www.kaggle.com/crowdflower/twitter-airline-sentiment>; SD4A-<https://www.kaggle.com/kamyab20/sentiment-dataset-for-afghanistan-sd4a>; Sentiment14-MV-<https://www.kaggle.com/kamyab20/sentiment140mv>; the code could be provided by request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdi, A.; Shamsuddin, S.M.; Hasan, S.; Piran, J. Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. *Inf. Process. Manag.* **2019**, *56*, 1245–1259. [CrossRef]
2. Chen, M.; Zhou, P.; Wu, D.; Hu, L.; Hassan, M.M.; Alamri, A. AI-Skin: Skin disease recognition based on self-learning and wide data collection through a closed-loop framework. *Inf. Fusion* **2020**, *54*, 1–9. [CrossRef]
3. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 1556–1566.
4. Er, M.J.; Zhang, Y.; Wang, N.; Pratama, M. Attention pooling-based convolutional neural network for sentence modelling. *Inf. Sci.* **2016**, *373*, 388–403. [CrossRef]
5. Liu, F.; Zheng, J.; Zheng, L.; Chen, C. Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification. *Neurocomputing* **2020**, *371*, 39–50. [CrossRef]
6. Xuanyuan, M.; Xiao, L.; Duan, M. Sentiment Classification Algorithm Based on Multi-Modal Social Media Text Information. *IEEE Access* **2021**, *9*, 33410–33418. [CrossRef]
7. Wang, X.; Liu, Y.C.; Sun, C.J.; Wang, B.X.; Wang, X.L. Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory. In Proceedings of the 53rd Annual Meeting of the Association-for-Computational-Linguistics (ACS)/7th International Joint Conference on Natural Language Processing of the Asian-Federation-of-Natural-Language-Processing (IJCNLP), Beijing, China, 26–31 July 2015; pp. 1343–1353.
8. Siddiqua, U.A.; Chy, A.; Aono, M. Tweet Stance Detection Using Multi-Kernel Convolution and Attentive LSTM Variants. *IEEE Trans. Inf. Syst.* **2019**, *E102D*, 2493–2503. [CrossRef]
9. Zhang, C.; Wang, D.; Wang, L.; Song, J.; Liu, S.; Li, J.; Guan, L.; Liu, Z.; Zhang, M. Temporal data-driven failure prognostics using BiGRU for optical networks. *J. Opt. Commun. Netw.* **2020**, *12*, 277–287. [CrossRef]

10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
11. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional Encoder Network for Targeted Sentiment Classification. *arXiv* **2019**, arXiv:1902.09314.
12. Usama, M.; Ahmad, B.; Song, E.; Hossain, M.S.; Alrashoud, M.; Muhammad, G. Attention-based sentiment analysis using convolutional and recurrent neural network. *Future Gener. Comput.-Syst. Int. J. eSci.* **2020**, *113*, 571–578. [\[CrossRef\]](#)
13. Rath, M.; Malik, A.; Varshney, D.; Sharma, R.; Mendiratta, S. Sentiment Analysis of Tweets Using Machine Learning Approach. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2–4 August 2018; pp. 1–3. [\[CrossRef\]](#)
14. Liu, Y.; Jiang, C.; Zhao, H. Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media. *Decis. Support Syst.* **2019**, *123*, 113079. [\[CrossRef\]](#)
15. Saeed, Z.; Ayaz Abbasi, R.; Razzak, I. EveSense: What can you sense from Twitter? In Proceedings of the 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, 14–17 April 2020; Volume 12036 LNCS, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer Science and Business Media Deutschland GmbH: Berlin/Heidelberg, Germany, 2020; pp. 491–495. [\[CrossRef\]](#)
16. Saeed, Z.; Ayaz Abbasi, R.; Razzak, M.I.; Xu, G. Event Detection in Twitter Stream Using Weighted Dynamic Heartbeat Graph Approach [Application Notes]. *IEEE Comput. Intell. Mag.* **2019**, *14*, 29–38. [\[CrossRef\]](#)
17. Oliveira, N.; Cortez, P.; Areal, N. Automatic creation of stock market lexicons for sentiment analysis using stocktwits data. In Proceedings of the 18th International Database Engineering and Applications Symposium, IDEAS 2014, Porto, Portugal, 7–9 July 2014; ACM International Conference Proceeding Series; Association for Computing Machinery: New York, NY, USA, 2014; pp. 115–123. [\[CrossRef\]](#)
18. Rasool, A.; Tao, R.; Kamyab, M.; Hayat, S. GAWA-A Feature Selection Method for Hybrid Sentiment Classification. *IEEE Access* **2020**, *8*, 191850–191861. [\[CrossRef\]](#)
19. Song, C.; Wang, X.K.; Cheng, P.F.; Wang, J.Q.; Li, L. SACPC: A framework based on probabilistic linguistic terms for short text sentiment analysis. *Knowl.-Based Syst.* **2020**, *194*, 105572. [\[CrossRef\]](#)
20. Sun, S.; Luo, C.; Chen, J. A review of natural language processing techniques for opinion mining systems. *Inf. Fusion* **2017**, *36*, 10–25. [\[CrossRef\]](#)
21. Arun, C.; Karthick, S.; Selvakumarasamy, S.; Joseph James, S. Car parking location tracking, routing and occupancy monitoring system using cloud infrastructure. *Mater. Today Proc.* **2021**. [\[CrossRef\]](#)
22. Rubtsova, Y. Automatic Term Extraction for Sentiment Classification of Dynamically Updated Text Collections into Three Classes. In *Knowledge Engineering and the Semantic Web*; Klinov, P., Mouromtsev, D., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 140–149.
23. Chen, X.; Tang, W.; Xu, H.; Hu, X. Double LDA: A Sentiment Analysis Model Based on Topic Model. In Proceedings of the 2014 10th International Conference on Semantics, Knowledge and Grids, Beijing, China, 25–29 August 2014; pp. 49–56. [\[CrossRef\]](#)
24. Fu, X.; Liu, W.; Xu, Y.; Cui, L. Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis. *Neurocomputing* **2017**, *241*, 18–27. [\[CrossRef\]](#)
25. Qin, P.; Xu, W.; Guo, J. An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing* **2016**, *190*, 1–9. [\[CrossRef\]](#)
26. Abid, F.; Alam, M.; Yasir, M.; Li, C. Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. *Future Gener. Comput.-Syst. Int. J. eSci.* **2019**, *95*, 292–308. [\[CrossRef\]](#)
27. Zhao, J.; Gui, X.; Zhang, X. Deep Convolution Neural Networks for Twitter Sentiment Analysis. *IEEE Access* **2018**, *6*, 23253–23260. [\[CrossRef\]](#)
28. Araque, O.; Corcuera-Platas, I.; Sánchez-Rada, J.F.; Iglesias, C.A. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst. Appl.* **2017**, *77*, 236–246. [\[CrossRef\]](#)
29. Kamkarhaghighi, M.; Makrehchi, M. Content Tree Word Embedding for document representation. *Expert Syst. Appl.* **2017**, *90*, 241–249. [\[CrossRef\]](#)
30. Zhao, R.; Mao, K. Fuzzy Bag-of-Words Model for Document Representation. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 794–804. [\[CrossRef\]](#)
31. Chen, G.; Xiao, L. Selecting publication keywords for domain analysis in bibliometrics: A comparison of three methods. *J. Inf.* **2016**, *10*, 212–223. [\[CrossRef\]](#)
32. Hu, K.; Wu, H.; Qi, K.; Yu, J.; Yang, S.; Yu, T.; Zheng, J.; Liu, B. A domain keyword analysis approach extending Term Frequency-Keyword Active Index with Google Word2Vec model. *Scientometrics* **2018**, *114*, 1031–1068. [\[CrossRef\]](#)
33. Stelzer, F.; Röhm, A.; Vicente, R.; Fischer, I.; Yanchuk, S. Deep neural networks using a single neuron: Folded-in-time architecture using feedback-modulated delay loops. *Nat. Commun.* **2021**, *12*, 5164. [\[CrossRef\]](#)
34. Dang, N.C.; Moreno-Garcia, M.N.; De la Prieta, F. Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics* **2020**, *9*, 483. [\[CrossRef\]](#)
35. Johnson, R.; Zhang, T. Deep pyramid convolutional neural networks for text categorization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2017; Volume 1, pp. 562–570. [\[CrossRef\]](#)
36. Rezaeinia, S.M.; Rahmani, R.; Ghodsi, A.; Veisi, H. Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **2019**, *117*, 139–147. [\[CrossRef\]](#)

37. Dos Santos, C.N.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014; pp. 69–78.
38. Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; pp. 225–230. [\[CrossRef\]](#)
39. Yoon, J.; Kim, H. Multi-channel lexicon integrated CNN-BILSTM models for sentiment analysis. In Proceedings of the 29th Conference on Computational Linguistics and Speech Processing, ROCLING 2017, Taipei, Taiwan, 27–28 November 2017; pp. 244–253.
40. Nguyen, H.T.; Nguyen, M.L. An ensemble method with sentiment features and clustering support. *Neurocomputing* **2019**, *370*, 155–165. [\[CrossRef\]](#)
41. Chatterjee, A.; Gupta, U.; Chinnakotla, M.K.; Srikanth, R.; Galley, M.; Agrawal, P. Understanding Emotions in Text Using Deep Learning and Big Data. *Comput. Hum. Behav.* **2019**, *93*, 309–317. [\[CrossRef\]](#)
42. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [\[CrossRef\]](#)
43. Wen, S.; Li, J. Recurrent convolutional neural network with attention for twitter and yelp sentiment classification arc model for sentiment classification. In Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, ACAI 2018, Sanya, China, 21–23 December 2018; The Hong Kong Polytechnic University: Hong Kong, China, 2018. [\[CrossRef\]](#)
44. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharrya, U.R. ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [\[CrossRef\]](#)
45. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [\[CrossRef\]](#)
46. Kumawat, S.; Yadav, I.; Pahal, N.; Goel, D. Sentiment Analysis Using Language Models: A Study. In Proceedings of the 11th International Conference on Cloud Computing, Data Science and Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 984–988. [\[CrossRef\]](#)
47. Sun, C.; Huang, L.; Qiu, X. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. *arXiv* **2019**, arXiv:1903.09588.
48. Wang, S.; Fang, H.; Khabsa, M.; Mao, H.; Ma, H. Entailment as Few-Shot Learner. *arXiv* **2021**, arXiv:2104.14690.
49. Onan, A.; Tocoglu, M.A. A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification. *IEEE Access* **2021**, *9*, 7701–7722. [\[CrossRef\]](#)
50. Yu, L.C.; Wang, J.; Lai, K.R.; Zhang, X. Refining Word Embeddings Using Intensity Scores for Sentiment Analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 671–681. [\[CrossRef\]](#)
51. Go, A.; Bhayani, R.; Huang, L. Twitter Sentiment Classification Using Distant Supervision. CS224N Project Report, Stanford. 2009. Available online: <https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf> (accessed on 20 October 2021).
52. Kamyab, M.; Tao, R.; Mohammadi, M.H.; Rasool, A. Sentiment analysis on Twitter: A text mining approach to the Afghanistan status reviews. In Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality, AIVR 2018, Nagoya, Japan, 23–25 November 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 14–19. [\[CrossRef\]](#)
53. Socher, R.; Perelygin, A.; Wu, J.Y.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Seattle, WA, USA, 18–21 October 2013; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2013; pp. 1631–1642.
54. Subba, B.; Kumari, S. A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Comput. Intell.* **2021**, Early Access. [\[CrossRef\]](#)
55. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2016; pp. 1480–1489.