JTE HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Deep Transform Ensemble Model for Sentiment Analysis

Quang Khai Tran, Huyen Trang Phan*
Ho Chi Minh City University of Technology and Education, Vietnam

*Corresponding author. *Email: trangpth@hcmute.edu.vn*

ARTICLE INFO

Received: 06/05/2025
Revised: 08/06/2025
Accepted: 16/06/2025
Published: 28/08/2025

KEYWORDS

BERT-CNN;
BERT-BiLSTM;
Deep transform ensemble;
Ensemble model;
Sentiment analysis.

ABSTRACT

The ensemble method is a technique that has garnered significant attention in recent years, particularly in the field of sentiment analysis. It leverages the strengths of multiple models to enhance overall performance. Although many ensemble methods for sentiment analysis have been proposed, few have incorporated deep learning models. In this study, we propose an ensemble model based on transformers and deep learning to improve sentiment analysis performance. The proposed model comprises the following main components: (i) an embedding layer, which converts input sentences into vector matrices; (ii) a BERT-LSTM-based sentiment classifier, which extracts and learns global and contextual features from the embedding layer; (iii) a BERT-CNN-based sentiment classifier, which extracts and learns local and semantic features from the embedding layer; (iv) an ensemble layer, which combines the extracted features; and (v) an ensemble classifier layer, which classifies the sentiment of the input sentences. The model is evaluated on four benchmark datasets. Experimental results show that it improves sentiment analysis performance by at least 0.02 and up to 0.05.

Doi: https://doi.org/10.54644/jte.2025.1897

1. Introduction

Sentiment analysis (SA) is the process of determining the level of sentiment expressed by users in their opinions about a particular topic or entity. Opinions can be images, audio, videos, and most commonly, texts. Sentiment level can be determined by sentiment score or by polarities such as positive, negative, neutral. SA is increasingly used in practical applications such as recommendation systems, virtual assistant systems, expert systems as a module that helps the system grasp the psychology and emotions of users, thereby providing more friendly and appropriate assistance and recommendations. Thus, the performance of SA methods directly affects the quality of these systems. Therefore, improving the performance of SA methods has been of interest to many researchers in recent years.

Table 1. *Survey state-of-the-art ensemble-based* SA *methods [5].*

Order	Ref	Year	Ensemble	Performance on IMDb dataset
1	[1]	2020	LR+SVM+RF	0.89
2	[2]	2021	CNN-LSTM	0.86
3	[3]	2019	CNN-BiLSTM	0.86
4	[4]	2022	BERT-LSTM	0.92
5	[6]	2021	LSTM	0.85
6	[7]	2020	BiLSTM	0.86
7	[8]	2019	CNN-BiLSTM	0.86

Many SA methods based on different approaches have been proposed, notably lexicon-based SA, machine learning-based SA, deep learning-based SA, and ensemble models-based SA [15]. Among them, SA based on ensemble models are achieving promising accuracy. Ensemble is a way of combining

individual models to solve the same problem. For example, combining machine learning models, combining deep learning models, combining lexicon-based models and machine learning models, deep learning. The purpose of ensemble is to take advantage of the strengths of each model. There are many SA methods based on ensemble models that have been proposed as summarized in Table 1.

From Table 1, we have some observations as follows: (i) An ensemble of machine learning, deep learning, and Transformers is the most popular combination method for SA; (ii) The ensemble methods involving BERT achieve better performance, with BERT being the most notable; (iii) No ensemble method combines BERT-CNN and BERT-BiLSTM for SA.

From the above observations, we propose an ensemble method for SA by combining BERT-CNN and BERT-BiLSTM, called Deep Transform Ensemble. The reason why we chose the two models, BERT-CNN and BERT-BiLSTM, to combine in this study is that: BERT uses BERT's character-level Byte Pair Encoding and static masking, allowing for representations with features of long dependencies and more semantics; meanwhile, CNN can well capture local features from the input sentence and BiLSTM can well capture global and contextual features. Therefore, combining BERT, CNN, and BiLSTM can extract more significant features. The Deep Transform Ensemble method includes the following main layers: (i) Embeddings layer is to convert input sentences into the vector space; (ii) BERT-LSTM based sentiment classifier that extracts and learns global and contextual features from embeddings layer; (iii) BERT-CNN based sentiment classifier that extracts and learns local and semantics features from embeddings layer; (iv) Ensemble layer that classifies sentiment of input sentences by integrating the outputs of BERT-LSTM and BERT-CNN layers into the final representation; (vi) Ensemble classifier is to compute distribution between sentiment classes for SA by using activation function. The Deep transform ensemble method was tested on four benchmark datasets, and the results demonstrated its performance. Our contributions are:

- We propose a Deep transform ensemble model to improve the performance of SA.

- We experiment with the Deep transform ensemble model on three datasets to evaluate the performance of the proposed model and ablations; data and code are available for a requirement.

The rest of the paper is organized as follows: Section 2 reviews related studies on the steps, their advantages, and disadvantages. Section 3 presents the research problem that the paper plans to address. Section 4 presents the detailed steps for building a deep transform ensemble model to improve the SA performance. Section 6 describes the experimental steps for the proposed model. Section 6 summarizes the objectives, steps, advantages, and disadvantages of the deep transform ensemble method for SA.

2. Related Works

Several recent studies have proposed hybrid approaches to leverage the strengths of different models. For example, Rehman et al. [9] introduced a hybrid architecture that combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network to improve the accuracy of SA on movie reviews. This approach exploits the CNN’s ability to extract local textual features and the LSTM’s strength in capturing long-term dependencies within word sequences. In this method, word embeddings are first generated using the Word2Vec model. These embeddings are then processed by the LSTM network to capture deeper semantic relationships and long-term dependencies. Finally, a deep CNN is applied to refine the embeddings further, generating multiple feature representations through various convolutional filters with different window sizes. The CNN effectively captures local features, while the LSTM addresses temporal dependencies and helps mitigate the vanishing gradient problem common in traditional recurrent neural networks (RNNs). This combination achieves competitive results on datasets such as IMDB and Amazon movie reviews. However, CNNs may require multiple convolutional layers to effectively model long-term dependencies, and their performance can degrade as the input sequence length increases.

Tan et al. [4] propose a model that combines RoBERTa (a Robustly Optimized BERT Pretraining Approach), a Transformer-based architecture, with Long Short-Term Memory (LSTM) networks for SA. The model leverages the Transformer’s ability to generate rich word embeddings and the LSTM’s capability to capture long-term contextual semantics, while also aiming to reduce the execution time constraints typically associated with regression-based models. The approach involves several stages,

including text preprocessing and data augmentation. Pre-trained RoBERTa weights are used to convert input text into meaningful embedding representations, which are then passed to an LSTM network to extract critical semantic features and long-range dependencies for sentiment classification. The model achieves high F1-scores on benchmark datasets such as IMDb, Twitter US Airline Sentiment, and Sentiment140. However, when used independently, regression models may suffer from longer execution times due to their inherently sequential processing nature. Additionally, prior to this study, Transformer-based models had not been extensively explored for this specific task. Additionally, the research group led by Kian Long Tan et al. [5] proposed a hybrid deep learning ensemble model for SA. This ensemble consists of three hybrid deep learning models, which combine RoBERTa, LSTM, BiLSTM, and GRU (Gated Recurrent Unit). The goal is to enhance overall performance by integrating the predictions of multiple models and addressing the issue of imbalanced data. The model achieves high accuracy on benchmark datasets such as IMDb, Twitter US Airline Sentiment, and Sentiment140. However, one limitation of this approach is its reliance on word frequency-based features, which can result in a loss of contextual interpretation.

Araque et al. [10] explored the combination of deep learning techniques with traditional machine learning methods for SA through ensemble techniques. They proposed a classification system that combines ensembles of surface features with deep learning features. The study examines two types of ensembles: ensembles of classifiers (using fixed rules and meta-learning) and ensembles of features. To establish a baseline, they developed a deep learning model using word2vec or doc2vec with logistic regression. The goal was to determine whether deep learning benefits from being combined with surface methods and to compare the performance of various deep and surface learning ensembles. They also provided a framework to characterize existing approaches for SA by extracting information from common word embeddings and combining it with surface features. The method introduces a deep learning baseline model (MG) that computes word embeddings using word2vec for short texts (with convolutional functions) or doc2vec for longer texts. These embeddings are then transformed into fixed-size vectors and fed into a logistic regression model. Subsequently, ensemble classifiers (CEM) are used to combine the predictions of the base classifiers (including traditional machine learning or dictionary-based models) with the deep learning baseline model, utilizing fixed rules (e.g., majority voting) or meta-learning techniques. The study also explores combinations such as CEMSG (surface features + common word embeddings) and feature ensembles that combine surface features (e.g., SentiWordNet dictionary scores, counts of punctuation marks, all-caps words, long words) with common word embeddings (MSG), affective word embeddings (MGA), or all three (MSGA) before feeding them into the classifier. However, traditional methods often rely on manual feature engineering, which can be time-consuming. Additionally, Bag of Words (BOW) methods lose word order and syntactic structure. Dictionary-based methods require consistent, reliable dictionaries but face challenges related to the variability of sentiment words across domains, contexts, and languages. Furthermore, it is not always clear whether deep learning models outperform traditional methods in terms of generalization ability, particularly when compared to domain-specific models. Combining word embeddings with convolutional functions for long documents may not always lead to significant improvements in sentiment classification performance.

Minaee et al. [11] proposed a hybrid model combining CNN and Bidirectional LSTM (Bi-LSTM) neural network for SA to capture temporal information using Bi-LSTM and extract local structure using CNN to improve SA performance by representing each word in the reviews using Glove embedding representation, then feeding these embedding representations into both CNN and Bi-LSTM models to predict sentiment by averaging the predicted scores from LSTM and CNN to make the final prediction. This hybrid model shows an improvement in performance compared to using CNN or LSTM models alone. It achieves high accuracy compared to previous studies. The study also plans to jointly train LSTM and CNN models in the future to further improve their performance.

In summary, recent research in SA has focused on leveraging the strengths of both deep learning and traditional methods, as well as exploring hybrid architectures and ensemble techniques to achieve improved performance. Proposed methods typically aim to capture both

JTE
HCMUTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Education
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

local features and long-term dependencies in text, while also addressing challenges such as imbalanced data.

3. Research Problem

This paper proposes an ensemble model of deep learning and transform to improve SA performance. The proposed model highlights the Deep transform ensemble models' ability to extract features related to more diverse and context-specific features, long dependencies, and semantic features, respectively, for SA. Research problem is presented as follows:

Given a set of m sentences/document $S = \{s_i \,|i \in [1,m]\}$, where s_i including a set of n words $s_i = \{w_j \,|j \in [1,k]\}$ where w_j is a j-th word in s_i. This study considers the SA task as a multiple classification issue, wherein each sentence/document is assigned a single label representing one of the sentiment polarities (positive, negative, or neutral). To put it another way, let's say we have a sentence s_i and we use $P = \{p_i \,|i \in [1,m],\ p_i \in \{pos, neg, neu\}\}$ to represent the labels of the sentence sentiment. The value of p_i is identified as follows:

$$\begin{cases} p_i = pos, if\ s_i \text{ expresses a positive sentiment} \\ p_i = neg, if\ s_i \text{ expresses a negative sentiment} \\ p_i = neu, if\ s_i \text{ expresses a neutral sentiment} \end{cases}$$

With the above notations, the research problem of this paper is formulated as follows: Given a set of m sentences/document S, and labeled sentence vector P_L. The objective of this research is to propose a Deep transform ensemble model to predict the vector P_U of the unlabeled sentence.

4. Proposed Method

The Deep transform ensemble model includes the following main layers: (i) Embeddings layer is to convert input sentences into the vector space; (ii) BERT-LSTM based sentiment classifier that extracts and learns global and contextual features from embeddings layer; (iii) BERT-CNN based sentiment classifier that extracts and learns local and semantics features from embeddings layer; (iv) Ensemble layer that classifies sentiment of input sentences by integrating the outputs of BERT-LSTM and BERT-CNN layers into the final representation; (vi) Ensemble classifier is to compute distribution between sentiment classes for SA by using activation function. The Deep transform ensemble model was tested on three benchmark datasets, and the results demonstrated its performance. The workflow of Deep transform ensemble model is illustrated in Figure 1.

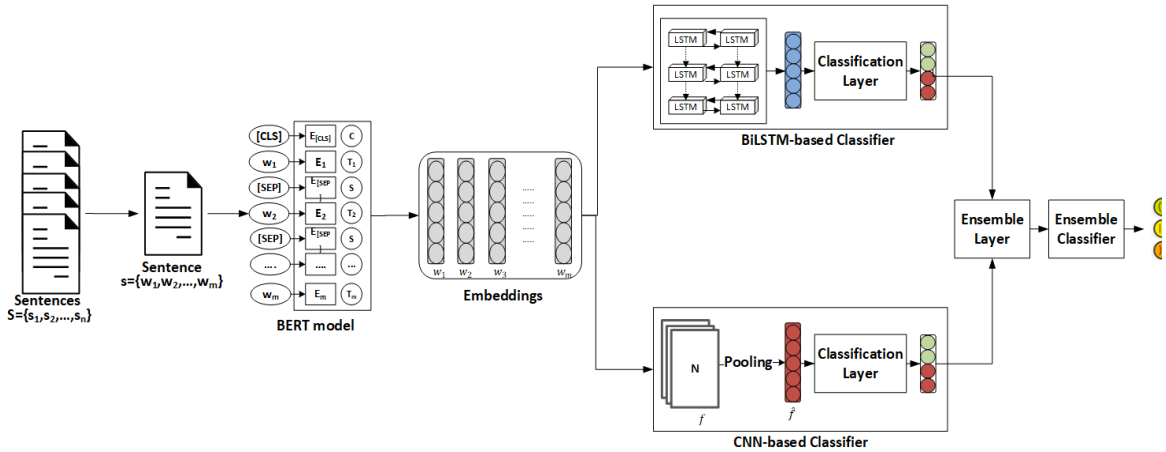

Figure 1. *Workflow of Deep transform ensemble model for SA.*

The following sections explain the steps used to construct the proposed model.

4.1. BERT Embeddings

This layer is to extract long dependencies semantics and contextual information at multiple levels of abstraction features in the sentence as follows:

$$O = BERT(s) \quad (1)$$

Where s is the input sentence, $BERT$ is the BERT model [12] and it includes the following layers:

Tokenizer layer: This layer aims to automatically insert some special token into the sentence, such as: a [CLS] token to mark starting of a sentence, a [SEP] token to mark ending of each sentence, a [PAD] symbol to mark a special token for padding, and a [UNK] mark a token not found in training set, to be easily convert words in the sentence into the vector space. This implies that the word sequence in the given sentence $s = \{w_1, w_2, w_3, w_4, w_5, \dots, w_n\}$ is converted into the form $\hat{s} = \{[CLS] + w_1, w_2 + [PAD] + w_3, w_4 + [UNK] + w_5, \dots, w_n + [SEP]\}$. This sequence of tokens is then fed into the embeddings layer.

Embeddings layer: This layer aims to convert the sequence of tokens into BERT embeddings that include three types of embeddings, such as word embedding, position embeddings, and token type embeddings.

Transformer layer: This layer is to generate the final representation vectors by using self-attention mechanism to combining word embedding, position embeddings, and token type embeddings.

Fully connected layer: This layer combines and transforms learned features from the final representation.

4.2. BERT-BiLSTM-based sentiment classifier

This classifier is to construct a sentiment classifier using a BiLSTM [13] on top of the BERT embeddings to extract contextual information at multiple levels of abstraction in the sentence as follows:

Input layer: The BERT embeddings are used as the input of this classifier.

BiLSTM layer: The BiLSTM is competent in learning the context information, which is consistent with the human language logic process that basic grammar depends on statistical characteristics and real meaning hidden between words depends mainly on the context. BiLSTM includes a forward LSTM to read the sentence from left to right and a backward LSTM to read the sentence from right to left. BiLSTM maps each word vector x_i to a pair of hidden vectors $\overrightarrow{h_\iota}$ and $\overleftarrow{h_\iota}$ as follows:

$$\overrightarrow{h_\iota} = \overrightarrow{lstm}\left(W_{o\vec{h}}o_i + W_{\vec{h}\vec{h}}\overrightarrow{h_{\iota-1}} + b_{\vec{h}}\right) \in R^{d_h}, i = [1, m] \quad (2)$$

$$\overleftarrow{h_\iota} = \overleftarrow{lstm}\left(W_{o\overleftarrow{h}}o_i + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{\iota+1}} + b_{\overleftarrow{h}}\right) \in R^{d_h}, i = [m, 1] \quad (3)$$

$$h_i = \left(W_{\vec{h}h}\overrightarrow{h_\iota} + W_{\overleftarrow{h}h}\overleftarrow{h_\iota} + b_h\right) \quad (4)$$

where $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$ are the forward and backward LSTM, respectively; W is the learnable matrix; for example, $W_{o\overleftarrow{h}}$ is the learnable matrix between the input vector and backward hidden vectors; $\overrightarrow{h_\iota}$ and $\overleftarrow{h_\iota}$ are the hidden states of $\overrightarrow{lstm}$ and $\overleftarrow{lstm}$, respectively; $\overleftarrow{h}_{i+1}$ is the next hidden vector of $\overleftarrow{h_\iota}$ and $\overleftarrow{h}_{i+1} = 0$; $\vec{h}_{i-1}$is the previous hidden vector of $\overrightarrow{h_\iota}$ and $\overrightarrow{h_0} = 0$.

Therefore, the contextualized word vector matrix $H = (h_1, h_2, \dots, h_m) \in R^{m \times d_h}$ is created from the matrix $0 \in R^{m \times d_w}$ where $h_i = [\overrightarrow{h_\iota}, \overleftarrow{h_\iota}]$ and d_h is the dimension of the feature vector.

Mean pooling layer: the contextualized word vector matrix H is then fed into the mean pooling layer to create the final feature vector as follows:

$$X = \frac{1}{m}\sum_{i=1}^{m} h_i \quad (5)$$

Fully connected layer: this layer is to adjust the sentiment characteristic of the previous layer and classifies sentiment polarity by using the activation function as follows:

$$y_B = Softmax(W_B.X + b_B) \quad (6)$$

where W_b is a learnable matrix of the activation function.

4.3. BERT-CNN-based sentiment classifier

This classifier is to construct a sentiment classifier using CNN model [14] on top of BERT embeddings to capture the local and semantics in the sentence as follows:

Convolutional layer: This layer aims to create a feature map, denoted by f, from the node embedding layer. The feature map is created by using a filter $N \in R^{j \times d}$ of length j from i to $i + j - 1$ to slide and filter important features. Each time sliding of the filter creates a new feature vector. The feature map is generated as follows:

$$f = [f_1, f_2, \dots, f_j] \tag{7}$$

Where $f_i = ReLU(N.O_{i:i+j-1} + b_C)$ where $ReLU$ is a rectified linear unit function; b_C is a bias term; O is the BERT Embeddings.

Max-pooling layer: this layer is to reduce the dimension of the feature map by taking the max value $\hat{f}_i = Max(f_i)$ as the feature corresponding to each filter. Assume that we use j filters; after this step, the obtained new feature is $\hat{f} = [\hat{f}_1, \hat{f}_2, \dots, \hat{f}_j]$. Then, this vector is fed into the next layer.

Fully connected layer: this layer is to adjust the sentiment characteristic of the previous layer and predicts sentiment polarity by using the Softmax function as follows:

$$y_C = Softmax(W_C.\hat{c} + b_C) \tag{8}$$

Where W_C is a learnable matrix of the Softmax function.

4.4. BERT-BiLSTM-CNN ensemble Classifier

This classifier is designed to combine outputs of BERT-BiLSTM and BERT-CNN classifiers to improve the performance of SA. This classifier includes the following steps.

Input layer: Two vectors y_B and y_C created from the BERT-CNN classifier and the BERT-BiLSTM classifier are fed into the ensemble classifier as the inputs.

Decision fusion layer: decision fusion is more potent since it merges data sources at the ending of the processing. In this study, decision fusion combines two output vectors of BERT-LSTM and BERT-CNN as follows:

$$g = concatenate(y_B, y_C) \tag{9}$$

where $concatenate$ is a combination operation, g is the final vector using for sentiment classification.

Ensemble classifier: The final vector is fed into the ensemble classifier layer using the softmax function to calculate the distribution value of the sentence's sentiment label as follows:

$$p = softmax(W_g \cdot g + b_g) \tag{10}$$

where W_g and b_g are a learnable matrix and a bias of the softmax function.

Model training: The deep transform ensemble model is trained by minimizing the cross-entropy error of the predicted and real sentiment label distributions as follows:

$$L = -\sum_i p_i log\hat{p}_i \tag{11}$$

where p is the real sentiment label distribution value of the sentence, and $\hat{p}$ is the predicted sentiment label distribution value of the sentence.

5. Experiment Results

5.1. Dataset

To prove the performance of the ensemble model, we experiment it on four benchmark datasets, such as Yelp, IMDb, amazon [8][1], and M2SA [15]. The detail of these four datasets are shown in Table 3 as follows:

Table 3. *Experimental datasets.*

Dataset	Total	No of labels	Training	Validation	Testing
Yelp	1000	2	800	100	100
IMDb	748	2	598	75	75
Amazon	1000	2	800	100	100
M2SA	15835	3	12668	1583	1584

5.2. Experimental Setting

The proposed method is trained on GPU 12GB (NVIDIA GeForce GTX 1080i), RAM 24GB, CPU (Intel® Core™ i7-7700CPU@3.60GHz), Samsung SSD 870 EVO 2TB. For parameter setting, a grid search is used to set up them for additional elements. The total number of trainable parameters for our proposal model for SA is illustrated in Table 4.

Table 4. *Hyperparameters of the proposed method.*

Key	BERT-BiLSTM-CNN ensemble
Word embeddings	768
Position embeddings	512
Token type embeddings	2
Batch size	16
No of BERT layer	12
Drop out	0.1
No of filters	100
Filter sizes	2, 3, 4
Hidden size	128
Activation	Softmax
Optimization	AdamW
Learning rare	2e-5

Evaluation metrics: On the aforementioned dataset, we employed accuracy as the statistic to assess and contrast the effectiveness of our proposal. By contrasting the actual and anticipated test set values, accuracy is determined.

5.3. Results and Discussions

In this paper, to prove the performance of the proposed method, we evaluate the results in the following cases: the general results, the ablation results, and the comparison results.

[1] https://www.kaggle.com/datasets/marklvl/sentiment-labelled-sentences-data-set

General results: The results of the proposed method are shown in Tables 5 and 6.

Table 5. *Training and validation performance of the proposed model.*

Epoch	Yelp				IMDb			
	T-Loss	**T-Acc**	**V-Loss**	**V-Acc**	**T-Loss**	**T-Acc**	**V-Loss**	**V-Acc**
1	0.70	0.72	0.49	0.82	0.78	0.68	0.50	0.88
2	0.40	0.85	0.41	0.84	0.47	0.85	0.40	0.91
3	0.34	0.88	0.37	0.84	0.41	0.86	0.35	0.96
4	0.32	0.88	0.36	0.86	0.37	0.89	0.30	0.96
5	0.30	0.89	0.35	0.85	0.33	0.91	0.28	0.96
6	0.28	0.91	0.34	0.86	0.33	0.90	0.26	0.97
7	0.28	0.90	0.33	0.87	0.31	0.92	0.25	0.96
8	0.27	0.91	0.33	0.86	0.31	0.91	0.25	0.97
9	0.26	0.91	0.32	0.86	0.31	0.91	0.24	0.97
10	0.27	0.90	0.32	0.86	0.31	0.90	0.24	0.97
Epoch	**Amazon**				**M2SA**			
	T-Loss	**T-Acc**	**V-Loss**	**V-Acc**	**T-Loss**	**T-Acc**	**V-Loss**	**V-Acc**
1	0.69	0.71	0.48	0.89	0.78	0.69	0.61	0.74
2	0.42	0.86	0.42	0.88	0.58	0.79	0.50	0.81
3	0.35	0.89	0.41	0.89	0.50	0.82	0.43	0.84
4	0.32	0.88	0.40	0.89	0.45	0.85	0.38	0.86
5	0.30	0.91	0.40	0.89	0.41	0.86	0.35	0.88
6	0.30	0.90	0.39	0.89	0.38	0.87	0.32	0.89
7	0.29	0.92	0.39	0.89	0.36	0.88	0.30	0.89
8	0.28	0.91	0.39	0.89	0.34	0.88	0.29	0.90
9	0.28	0.90	0.39	0.89	0.33	0.89	0.29	0.90
10	0.28	0.90	0.39	0.89	0.33	0.89	0.29	0.90

Table 6. *Average training and validation performance of the proposed model.*

Method	**T-Loss**	**T-Acc**	**V-Loss**	**T-Loss**	**T-Time**
IMDb	0.39±0.07	0.87±0.04	0.31±0.06	0.95±0.03	32.1s
Yelp	0.34±0.01	0.88±0.005	0.36±0.03	0.85±0.01	44.2s
Amazon	0.35±0.004	0.88±0.006	0.41±0.004	0.89±0.002	43.0s
M2SA	0.45±0.003	0.84±0.002	0.38±0.003	0.86±0.005	12min41.3s

Table 7. *Testing performance of the proposed method.*

Class	Yelp				IMDb			
	Precision	**Recall**	**F1 Score**	**Support**	**Precision**	**Recall**	**F1 Score**	**Support**
0	0.75	0.87	0.81	46	0.84	0.91	0.87	34
1	0.87	0.76	0.81	54	0.92	0.85	0.89	41

Accuracy	0.81±0.03				0.88±0.01			
Loss	0.37±0.005				0.36±0.001			
Class	**Amazon**				**M2SA**			
	Precision	**Recall**	**F1 Score**	**Support**	**Precision**	**Recall**	**F1 Score**	**Support**
0	0.87	0.92	0.89	36	0.79	0.84	0.81	387
1	0.95	0.92	0.94	64	0.98	0.98	0.98	657
2					0.88	0.85	0.86	540
Accuracy	0.92±0.01				0.91±0.005			
Loss	0.25±0.006				0.28±0.002			

Figure 2. *The performance of the proposed method.*

Looking at Tables 5, 6 and 7, and Figure 2, we can see that the proposed deep transform ensemble method has achieved the expected performance in terms of loss and accuracy for all training, validation, and testing. This shows that the Deep transform ensemble method performs quite well for classification tasks, including binary and multiple. However, among them, the performance achieved on the Amazon and M2SA datasets is better than on the Yelp and IMDb datasets. Specifically, on the Amazon and M2SA datasets, testing accuracy is higher than training and validation accuracy, and testing loss is lower than training and validation loss. Meanwhile, on the Yelp and IMDb datasets, testing accuracy is not higher than training and validation accuracy, and testing loss is not better than training and validation loss. The main reason is that the number of sentences in the Amazon and M2SA datasets is larger than the IMDb dataset. Besides, although the Yelp dataset has the same number of sentences as the Amazon dataset, performance is not as good because the data on the Yelp dataset contains more noise.

Ablation results: To confirm the necessity of all component models used to build the deep transform ensemble method, we experiment the proposed method by implementing ablations as follows: (i) No CNN indicates the ablation method by ignoring the CNN layer. (ii) No BiLSTM indicates the ablation method that ignores the BiLSTM layer. The performance of ablations is shown in Table 8.

Table 8. *Ablation performance.*

			Yelp			
Method	T-Loss	T-Acc	V-Loss	V-Acc	Te-Loss	Te- Acc
No CNN	0.47	0.82	0.50	0.83	0.42	**0.86**
No BiLSTM	**0.33**	**0.88**	**0.36**	**0.85**	**0.37**	0.79
Ensemble model	0.34±0.01	**0.88**±0.005	**0.36**±0.03	**0.85**±0.01	**0.37**±0.005	0.81±0.03
			IMDb			
Method	T-Loss	T-Acc	V-Loss	V-Acc	Te-Loss	Te- Acc
No CNN	0.56	0.77	0.54	0.79	0.53	0.76

No BiLSTM	**0.37**	**0.87**	**0.31**	**0.95**	**0.34**	**0.88**
Ensemble model	0.39±0.07	**0.87**±0.04	**0.31**±0.06	**0.95**±0.03	0.36±0.001	**0.88**±0.01
			Amazon			
Method	T-Loss	T-Acc	V-Loss	V-Acc	Te-Loss	Te- Acc
No CNN	0.49	0.8	0.51	0.81	0.38	0.85
No BiLSTM	**0.34**	**0.88**	**0.41**	0.88	**0.25**	0.90
Ensemble model	0.35±0.004	**0.88**±0.006	**0.41**±0.004	**0.89**±0.002	**0.25**±0.006	**0.92**±0.01
			M2SA			
Method	T-Loss	T-Acc	V-Loss	V-Acc	Te-Loss	Te- Acc
No CNN	0.58	0.77	0.51	0.80	0.38	0.85
No BiLSTM	**0.44**	**0.85**	**0.37**	**0.87**	0.29	**0.90**
Ensemble model	0.45±0.003	0.84±0.002	**0.38**±0.003	0.86±0.005	**0.28**±0.002	**0.91**±0.005

Looking at Table 8, we can see that the deep transform ensemble method still achieves the best loss and accuracy for training, testing, and validation. In particular, the difference in loss and accuracy between the deep transform ensemble method and the No CNN method is quite high. This shows that the CNN layer plays a quite important role in extracting local and semantic features. Meanwhile, the difference between the deep transform ensemble and the No BiLSTM method is insignificant. This shows that the BiLSTM layer in ensemble-based SA does not capture global and contextual features well.

Comparison results: To confirm the deep transform ensemble method can improve the performance of SA method, we compare its performance in terms of accuracy, precision, recall, and F1 score with the baselines such as LSTM [6], BiLSTM [7], CNN-LSTM [2], CNN-BiLSTM [3]. In this comparison, we admit the performance that published by authors in previous publications. The compared performance is listed in Table 8.

Table 9. *Comparison performance on IMDb dataset.*

Method	**Accuracy**	**Precision**	**Recall**	**F1 score**
LSTM [6]	0.85	0.85	0.85	0.85
BiLSTM [7]	0.86	0.87	0.87	0.86
CNN-LSTM [2]	0.86	0.86	0.86	0.86
CNN-BiLSTM [3]	0.86	0.86	0.86	0.86
Deep transform ensemble	**0.88**	**0.88**	**0.88**	**0.88**

Looking at Table 9, we see that the performance of our deep transform ensemble for SA is best in terms of four metrics by at least 0.02 and up to 0.05. That confirms the combination of BERT, CNN, and BiLSTM by ensembling them, which takes advantage of the ensembled models individually to extract features for SA.

6. Conclusions

This research introduces an ensemble model from BERT, CNN, and BiLSTM, called deep transform ensemble, aiming to take advantage of the combined models. Specifically, it takes advantage of the ability to extract the semantics features from BERT, the local features from CNN, and the global and long dependency features from BiLSTM. The proposed deep transform ensemble model includes five main parts: Embeddings layer, BERT-CNN layer, BERT-BiLSTM layer, Ensemble layer, and Ensemble classifier. The proposed method is experimented on four benchmark datasets consisting of binary and multiple classifications. The experimental results prove the performance of the deep transform ensemble

model in terms of accuracy and loss compared with the baselines. Although the proposed method has improved the performance of some previous SA methods, it is based on ensemble techniques, making it difficult to avoid making the model architecture complex. In future research, we will be interested in lightweighting the models used for SA while maintaining their accuracy as a research direction.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] P. Meel, P. Chawla, S. Jain, and U. Rai, "Web text content credibility analysis using max voting and stacking ensemble classifiers," in *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, 2020, pp. 157-161.
[2] P. K. Jain, V. Saravanan, and R. Pamula, "A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1-15, 2021.
[3] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, "A CNN-BiLSTM model for document-level sentiment analysis," *Machine Learning and Knowledge Extraction*, vol. 1, no. 3, pp. 832-847, 2019.
[4] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21517-21525, 2022.
[5] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, "Sentiment analysis with ensemble hybrid deep learning model," *IEEE Access*, vol. 10, pp. 103694-103704, 2022.
[6] M. S. Hossen, A. H. Jony, T. Tabassum, M. T. Islam, M. M. Rahman, and T. Khatun, "Hotel review analysis for the prediction of business using deep learning approach," in *2021 international conference on artificial intelligence and smart systems (ICAIS)*, 2021.
[7] A. Garg and R. K. Kaliyar, "PSent20: An effective political sentiment analysis with deep learning using real-time social media tweets," in *2020 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE)*, 2020.
[8] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015.
[9] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, pp. 26597-26613, 2019.
[10] O. Araque, I. Corcuera-Platas, J. F. Snchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236-246, 2017.
[11] S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models," arXiv preprint arXiv:1904.04206, 2019.
[12] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2019.
[13] H. T. Phan, N. T. Nguyen, and D. Hwang, "Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis," *Information Sciences*, vol. 589, pp. 416-439, 2022.
[14] Y. Kim, "Convolutional neural networks for sentence classification. arXiv 2014," *arXiv preprint arXiv:1408.5882*, 2014.
[15] H. T. Phan, N. T. Nguyen, D. Hwang, and Y. S. Seo, "M2SA: A novel dataset for multi-level and multi-domain sentiment analysis," *Journal of Information and Telecommunication*, vol. 7, no. 4, pp. 494-512, 2023.

Quang Khai Tran received the master's degree in computer science from University of Information Technology – Vietnam National University, Ho Chi Minh City in 2019. Currently, he lectures at the Faculty of Information Technology within the Ho Chi Minh City University of Technology and Education in Vietnam. His research interests include computer vision, image processing and person re-identification. Email: khaitq@hcmute.edu.vn. ORCID: https://orcid.org/0009-0005-0804-6550

Huyen Trang Phan received the M.S. degree in computer science from the University of Science and Technology - The University of Da Nang, Vietnam, in 2015, Ph.D. degree and Postdoctoral in Computer Science at the Department of Computer Engineering from Yeungnam University, South Korea in 2020 and 2021. She worked as a research professor at the Department of Computer Engineering, Yeungnam University, South Korea, from 2021 to 2024. She is currently a lecturer at the Faculty of Information Technology, Ho Chi Minh City University of Technology and Education, Vietnam. She is the author of 10 journal papers and 15 conference papers. Her research interests include sentiment analysis, fake news detection, text summarization, and decision support systems.
Email: trangpth@hcmute.edu.vn. ORCID: https://orcid.org/0000-0002-7466-9562