JURNAL INFOTEL
VOL. 17, NO. 3, AUGUST 2025, PP. 564–583. DOI:10.20895/INFOTEL.V17I3.1396

RESEARCH ARTICLE

Attention Mechanisms on Hybrid Models: Examining the Impact of Sentence Length in Sentiment Analysis

Livia Naura Aqilla[1,*] and Yuliant Sibaroni[2]

[1,2]School of Computing, Telkom University, Bandung 40257, Indonesia

*Corresponding email: liviaaqilla@student.telkomuniversity.ac.id

Received: June 06, 2025; Revised: August 17, 2025; Accepted: August 18, 2025.

Abstract: Sentiment analysis is a key task in natural language processing (NLP) with applications in a wide range of domains. This study examines the impact of self-attention and global attention placement in CNN-BiLSTM and CNN-LSTM models, exploring their effectiveness when positioned before, after or both before and after BiLSTM/LSTM, particularly for texts of different lengths. Instead of applying attention mechanisms in a fixed position, this research explores the most suitable type and placement of attention to improve model understanding and adaptability across datasets with different text lengths. Experiments were conducted using the IMDB Movie Reviews Dataset and the Twitter US Airline Sentiment dataset. The results show that for long texts, CNN-BiLSTM with self-attention before and after BiLSTM achieves an F1 score of 93. 77% (+2.72%), while for short texts, it reaches 82.70% (+2.24%). These findings highlight that optimal attention placement significantly improves sentiment classification accuracy. The study provides insights into designing more effective hybrid deep learning models. It contributes to future research on multilingual and multi-domain sentiment analysis, where attention mechanisms can be adapted to different textual structures.

Keywords: Attention Mechanism, CNN BiLSTM, Hybrid Deep Learning, Sentiment Analysis, Text Classification

1 Introduction

Sentiment analysis, or opinion mining, is a research field that focuses on analyzing and extracting insights from subjective text, such as opinions about individuals, events and

 ISSN: 2085-3688; e-ISSN: 2460-0997

products [1], [2]. The development of deep learning methods for sentiment analysis has significantly improved classification accuracy; however, challenges remain, particularly in managing complex language structures, capturing long-range dependencies, and optimizing effective attention mechanisms [3]. One of the primary challenges in sentiment analysis is handling variations in text length. Short texts require rapid contextual understanding from limited information, whereas long texts demand that models maintain meaning and coherence throughout the document. These challenges are often difficult to overcome using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), underscoring the importance of evaluating how attention mechanisms can enhance a model's ability to handle text length variations [3].

In natural language processing (NLP), Convolutional Neural Networks (CNNs) are widely used to extract local features through convolution and pooling operations. In contrast, Recurrent Neural Networks (RNNs) excel in processing sequential data of varying length [4]. However, CNNs struggle to preserve word order and relationships, while RNNs suffer from the problem of vanishing gradients when processing long texts [5]. Long-Short-Term Memory (LSTM) networks were introduced to address this problem, incorporating a gating mechanism that allows the model to retain essential information over long sequences [6]. Bidirectional LSTM (BiLSTM) further enhances this approach by capturing information from both forward and backward directions within a sequence, making it effective for various text-based tasks [7]. Furthermore, effective word embeddings, such as those generated by RoBERTa [8], offer robust contextual representations that facilitate long-range dependency modeling in downstream layers.

Meanwhile, attention mechanisms have shown their effectiveness in multiple NLP tasks, including sentiment analysis, by enabling models to focus on important words [9] selectively [10]. In short texts, attention highlights key words that carry sentiment, whereas in long texts, it preserves contextual relationships throughout processing [9], [10]. Consequently, hybrid models that integrate these mechanisms offer significant potential for improving contextual understanding across different text lengths. Semary et al. [11] and Mohbey et al. [12] demonstrated that the integration of CNN-LSTM effectively handles short texts, such as tweets, outperforming conventional machine learning methods by using both spatial and sequential features.

In addition to CNN-LSTM, CNN-BiLSTM architectures with multi-attention mechanisms have proven effective in capturing hierarchical features and contextual information. Wankhade et al. [10] demonstrated that their model outperformed state-of-the-art approaches on film review datasets. In contrast, Li et al. [13] combined BERT, BiLSTM, CNN, and attention to enhance sentiment classification accuracy on the Weibo platform. Despite their effectiveness, transformer-based mechanisms, such as multihead attention, often introduce significant computational overhead, which is not always necessary in architectures like CNN-BiLSTM. Kardakis et al. [7] noted that self-attention and global attention can already achieve strong performance in sentiment tasks without added complexity.

Attention mechanisms have consistently been shown to enhance sentiment analysis performance by enabling models to focus on the most informative text segments. Xia et al. [14] demonstrated that self-attention is particularly effective in capturing distant word relationships, especially in long text datasets such as Twitter and IMDb. Kardakis et al. [7] compared self-attention, global attention, and hierarchical attention, showing that attention-based models can improve accuracy by up to 3.5%, with self-attention outperforming others in long sequence scenarios. These findings, based on the foundational work

of Vaswani et al. [15], highlight the ability of attention mechanisms to overcome the limitations of RNNs by selectively focusing on semantically relevant sentence segments.

In addition to improving performance, recent studies emphasize the strategic placement of attention within hybrid architectures. Wei et al. [16] found that the application of self-attention after the BiLSTM layers significantly improved the capture of long-range dependencies. Zeng et al. [17] introduced PosATT-LSTM, which utilizes position-aware attention to prioritize contextually relevant words, resulting in substantial gains in aspect-based sentiment classification. Similarly, Sifak and Setiawan [18] showed that attention, when optimally placed in a CNN-BiGRU model, improved the detection accuracy of hate speech.

Although numerous previous studies have confirmed the effectiveness of attention mechanisms in improving sentiment classification, the topic of attention placement - whether before, between, or after core components within hybrid architectures — remains relatively underexplored in the literature. To address this gap, this study investigates the impact of attention placement in CNN-BiLSTM and CNN-LSTM models by evaluating their performance across different text lengths using short text (Twitter US Airline) and long text (IMDB Movie Reviews) datasets. The findings are expected to provide deeper insight into optimizing attention-based hybrid models for sentiment classification in various textual contexts.

2 Research Method

The methodology phases are illustrated in Figure 1, which outlines the overall design system and how each stage contributes to achieving the research objectives.

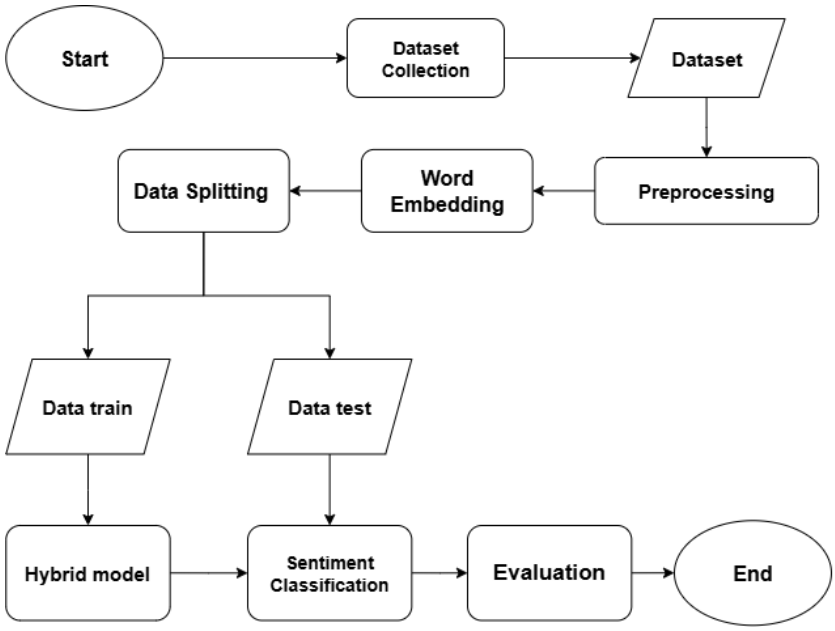

Figure 1: Flowchart design system.

2.1 Dataset

The author will utilize open-source datasets available on the internet for this research. The datasets used in this study are:

1. IMDB Dataset: This dataset contains 50,000 highly polar movie reviews categorized as positive or negative sentiments. It is widely used in sentiment analysis due to its balanced nature and comprehensive representation [19].
2. Twitter Airline Sentiment Dataset: This dataset consists of 14,640 tweets about US airlines with labeled sentiments as positive, negative, or neutral. It is particularly suitable for analyzing sentiment in social media contexts [20].

The IMDB dataset contains structured long-form movie reviews labeled positive or negative, while the Twitter US Airline dataset consists of short, informal, and predominantly negative posts. To address class imbalance in the latter, synonym replacement is employed as a data augmentation technique, increasing the representation of minority classes without altering semantic content [21]. This study evaluates the effectiveness of attention mechanisms in improving sentiment classification in diverse text lengths, structures, and linguistic styles.

2.2 Preprocessing

In sentiment analysis, data preprocessing plays a crucial role in eliminating extraneous elements that could negatively impact model performance [6]. Since most of the texts used in this study are tweets, special characters, numbers, and punctuation marks were removed to reduce noise and retain only relevant information [22]. Figure 2 illustrates the preprocessing steps [23].

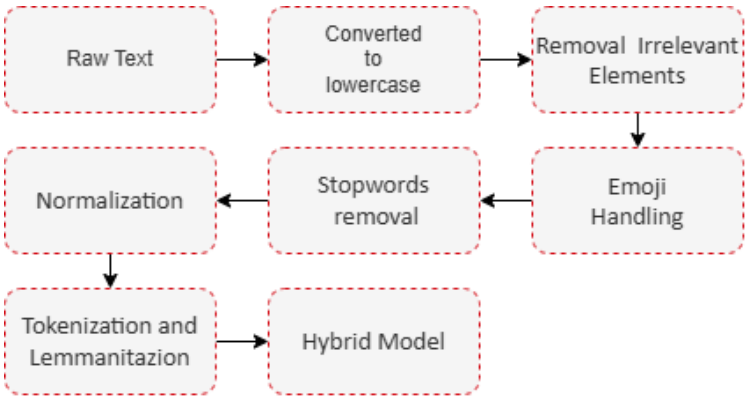

Figure 2: Data preprocessing steps.

- **Lowercasing:** All text was converted to lowercase to ensure consistency and avoid distinguishing identical words with different capitalizations
- **Removal of Irrelevant Elements:** Irrelevant components such as hashtags (e.g., topic), usernames (e.g., @username), and URLs (e.g., "http," “https,” “www”) were

removed to eliminate superfluous content that does not contribute to sentiment analysis.

- **Expansion of Contractions:** To preserve the meaning of contractions, contractions like "isn't" and "don't" were expanded into their full forms, such as "is not" and "do not."
- **Emoji Handlings:** ASCII emoticons (e.g., :), :D) were converted into textual representations (e.g., :) → ("smile") to ensure sentiment preservation. Similarly, Unicode emojis, which convey mood and emotion,were transformed into descriptive text using the demojize() method from Python's emoji package.
- **Stopword Removal:** Stopwords (e.g., "the," "and," "is") were removed to reduce noise, as they do not significantly impact sentiment analysis. However, negations (e.g., "not," "no") were preserved since they can alter sentiment meaning.
- **Normalization of Repeated Characters:** To ensure consistency, exaggerated words (e.g., "sooo happy") were standardized by limiting repeated characters (e.g., "soooo happy" → "soo happy").
- **Tokenization and Lemmatization:** applied to standardize textual data and improve feature extraction. Tokenization breaks sentences into individual tokens (words), while lemmatization reduces words to their base or dictionary form (e.g., "running" → "run"), promoting consistency across word variations.

2.3 Data Splitting

Data splitting divides the preprocessed dataset into training and testing subsets, using a 90:10 ratio for training and testing. It ensures effective model training and unbiased evaluation. The scikit-learn library is utilized for this process.

2.4 Hybrid Model Approach

The hybrid model, equipped with an attention mechanism, captures both local and global textual features. Figure 3 illustrates its architecture, integrating RoBERTa with CNN-BiLSTM and CNN-LSTM for sentiment classification. The model consists of the following components:

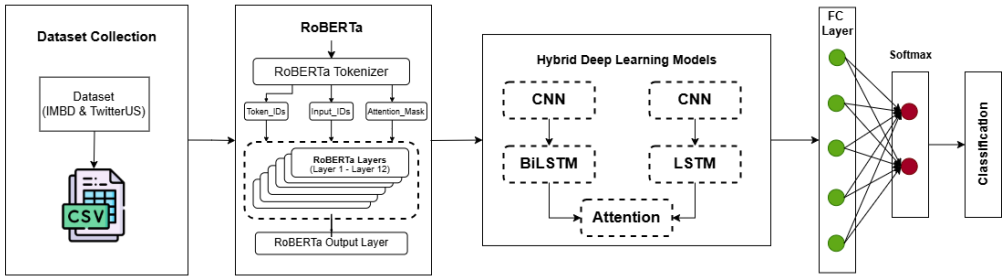

Figure 3: Illustration of hybrid models architecture.

2.4.1 Word Embedding Layer

In this research, RoBERTa (Robustly optimized BERT approach) is utilized as the primary mechanism to generate contextual word embeddings. RoBERTa is a transformer-based model that enhances the BERT architecture through optimized training strategies, including dynamic masking, removal of the next sentence prediction, and training on significantly larger datasets [24]. RoBERTa generates contextualized word embeddings for the input text. Given a sequence $X = \{x_1, x_2, ..., x_L\}$, RoBERTa outputs embeddings $E = \{e_1, e_2, ..., e_L\}$, where each embedding $e^d_{i \in R}$. RoBERTa generates embeddings by processing text through a transformer architecture that applies self-attention mechanisms to capture both semantic and syntactic nuances [25]. Unlike static embeddings such as Word2Vec or GloVe, RoBERTa offers richer text representations, which enhances the deep learning model's ability to comprehend semantics and word similarities [24]. The 'RoBERTa-base' pre-trained word embedding model is used, typically supporting a maximum of 512 tokens.

2.4.2 Convolutional Neural Network (CNN)

Deep learning algorithms are believed to produce excellent results in NLP [26]. CNNs are a type of neural network that is used primarily for visual data analysis, but can also be highly effective for text processing when combined with advanced embeddings [27]. In this model, we utilize a CNN layer to analyze the dense, context-aware embeddings produced by RoBERTa, effectively capturing the semantic intricacies of the text [27]. CNN maintains a hierarchical structure by learning internal feature representations and capturing local and global text patterns, enabling effective generalization in text classification tasks [28]. A convolutional Neural Network is a feed-forward neural network. It can provide important input information with fewer parameters for other deep learning models to use [29]. CNN has two layers, namely the convolutional layer and the max-pooling layer. CNN consists of an input, convolutional, pooling, fully connected, and output layer.

The CNN layer captures the local patterns in the text by applying a convolution operation. The calculation for the feature map is presented in Eq. (1).

$$\mathrm{F_{conv}} = \mathrm{ReLU}(\mathrm{W_{conv}} * \mathrm{E} + \mathrm{b_{conv}}) \tag{1}$$

where $\mathrm{W_{conv}}$ and $\mathrm{b_{conv}}$ are the learnable weights and biases. ReLU is used to make model training faster and more stable by overcoming vanishing gradients.

2.4.3 Bidirectional LSTM (BiLSTM)

BiLSTM combines forward Long Short-term Memory (LSTM) and backward LSTM [30]. BiLSTM is another type of Recurrent Neural Network (RNN). BiLSTM fully considers the context of the previous sentence and the next sentence to extract bidirectional information. The BiLSTM model exhibits greater predictive power than the Long Short-term Memory (LSTM) model [31]. In Long Short-term Memory (LSTM), the data is generated in a forward direction, indicating that the state at time t depends only on the data up to and including time t. Thus, the generated data are as vital as the previous one to describe the entire input semantics. For a better representation of the relevant data, the BiLSTM model is used. The result of the BiLSTM organization is determined by combining the consequences of the relevant results of the previous layer and the layer after each time point [32]. The BiLSTM

layer processes the data sequentially in the forward and backward directions, producing hidden states with the calculation in Eq. (2):

$$h_t = [h_t^{fwd}; h_t^{bwd}] \tag{2}$$

where h_t^{fwd} and h_t^{bwd} are the forward and backward hidden states.

2.4.4 Long short-term (LSTM)

Long Short-Term Memory (LSTM) is a specialized type of Recurrent Neural Network (RNN) introduced by Hochreiter and Schmidhuber to address the vanishing gradient problem in training sequential neural networks [33]. LSTM employs a memory cell and three main gates: the forget gate, input gate, and output gate. The memory cell allows LSTM to retain important information over extended time intervals, distinguishing it from standard RNNs. LSTM is designed to capture long-term dependencies in sequential data. The forget gate determines which information should be removed from the memory cell based on the current input and the previous state of the memory cell. The input gate decides what new information should be stored in the memory cell, while the output gate regulates the portion of the information from the memory cell to be used as output at time t [34]. At each time step t, the computations in an LSTM unit can be described as shown in Eq. (3) to Eq. (5).

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$
$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

The cell state and hidden state are then computed as follows (see Eq. (6) to Eq. (8)):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{6}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7}$$
$$h_t = o_t \odot \tanh(C_t) \tag{8}$$

where σ represents the sigmoid function, $\tanh$ is the hyperbolic tangent function, f, i, and o are the forget, input and output gates, respectively, while C and h denote the state of the memory cell and the hidden state at time t [2]. LSTM is widely applied in text analysis tasks, including sentiment analysis, due to its ability to capture temporal context and relationships between elements in sequences [3].

2.5 Attention Mechanism

Attention is a key concept in deep learning that has driven major advances in neural network performance [35]. It has been widely applied in NLP tasks such as sentiment analysis and opinion mining [7]. To address the challenges of handling texts of varying lengths, this study specifically evaluates two attention architectures with fundamentally different mechanisms: self-attention and global attention. The selection of these two mechanisms is based on how they process different contexts, which are hypothesized to have different impacts depending on text length.

The fundamental difference between self-attention and global attention lies in their scope and how it processes context. Self-attention works internally by mapping dependencies between elements within the same sequence. At the same time, global attention operates comprehensively by focusing on different parts of the entire input to create a concise representation. This difference in approach makes the internal focus of self-attention highly effective in capturing complex relationships in short texts. At the same time, the ability of global attention to comprehensively summarize information is more advantageous for analyzing long texts where important information is often scattered.

2.5.1 Self-Attention

Self-attention enables models to capture dependencies across an entire sequence by computing attention weights based on the relevance between elements [36]. This mechanism addresses the limitations of RNNs in modeling long-range dependencies.

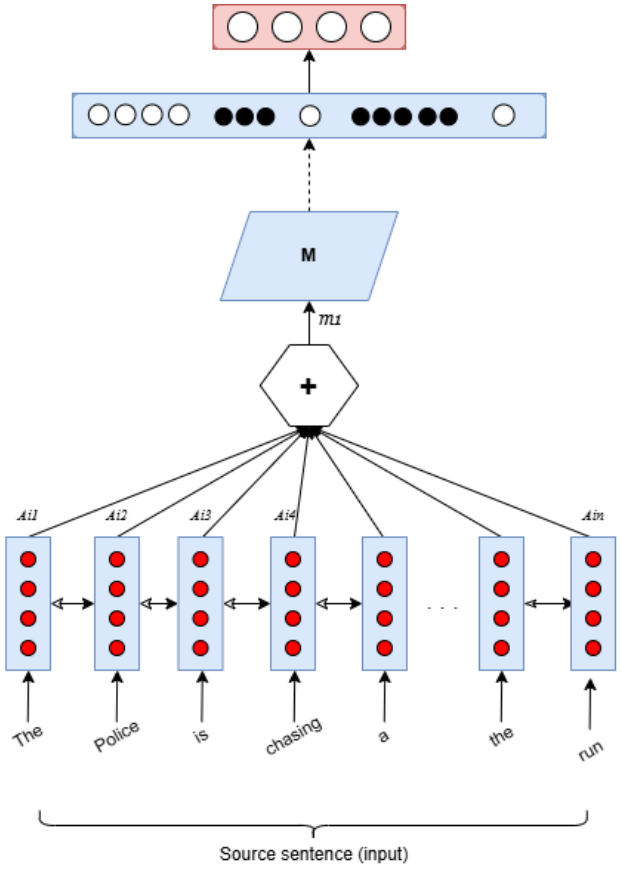

Figure 4: The structure of the self-attention mechanism.

Mathematically, self-attention is formulated as follows.

$$\mathrm{A} = \mathrm{softmax}(W_{s2}\tanh(W_{s1}H^T)) \tag{9}$$

$$\mathrm{M} = \mathrm{AH} \tag{10}$$

Where H is the matrix of hidden states produced by a BiLSTM, and W_{s1}, W_{s2} are learned weight matrices. The matrix A represents the attention weights that determine the importance of each sequence element, while M is the final sentence representation. $\tanh$ allows the model to learn more complex relationships between words. The combination of W_{s1}, $\tanh$, and W_{s2} works like a small 'brain' to calculate attention scores, resulting in more accurate results than simple linear calculations.

To prevent redundant attention distributions, a regularization term is introduced, as shown in Eq. (11):

$$P = \left\|AA^T - I\right\|_F^2 \tag{11}$$

where I is the identity matrix and $\|\cdot\|_F$ denotes the Frobenius norm. This term promotes diverse attention distributions.

The structure of the self-attention mechanism is illustrated in detail in Figure 4. The architecture workflow begins with the input sentence (*source sentence*), where a BiLSTM layer processes each token to generate a series of hidden state vectors. Each block in this layer represents one hidden state vector, with the red dots within visually symbolizing the dimensions or features that constitute the vector. Subsequently, the attention mechanism computes weights (denoted by A_{i1}, A_{i2}, $\cdots$) to determine the relevance of each token. These vectors are then aggregated through a weighted summation operation (symbolized by the hexagon '+') to form the final sentence representation, matrix M. This matrix M, a contextual summary of the entire sentence, is then passed to a fully connected layer (the blue rectangle) for further feature transformation before proceeding to the output layer (the red rectangle), which produces the final prediction of the model.

2.5.2 Global Attention

Global attention allows the decoder to dynamically focus on different parts of the input sequence at each decoding step, rather than relying on a fixed length context vector [37]. Figure 5 provides a visual breakdown of the components of the global attention mechanism. The blue blocks at the bottom represent the hidden states of the encoder ($\bar{h}_s$), corresponding to each token in the source sequence, while the red blocks represent the hidden states of the decoder (h_t). The red dots within these blocks visually depict the dimensions or features that constitute each hidden state vector. The Attention Layer (enclosed in the dashed box) computes the alignment weights (a_t) by comparing the current decoder state (h_t) with all encoder states. These weights are used to create a weighted average of the encoder states, resulting in the Context Vector (c_t). Finally, this Context Vector is combined with the current decoder state to produce the final hidden state (the gray block), which is used to generate the output token y_t.

The attention weights in global attention are computed as follows:

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{L}\exp(e_{ik})} \tag{12}$$

where the alignment score e_{ij} is calculated as:

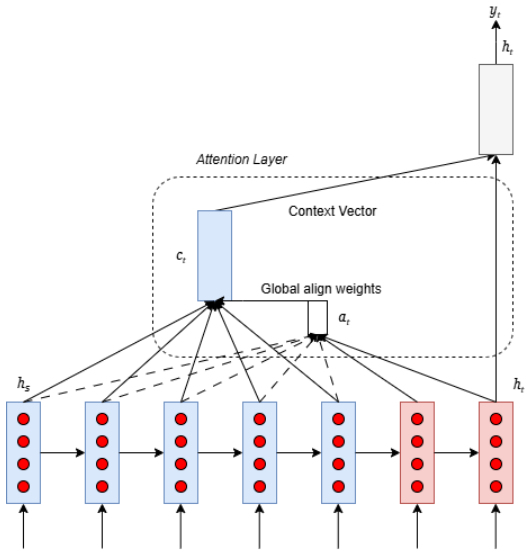

Figure 5: Illustration of the global attention mechanism.

$$e_{ij} = a(s_{i-1}, h_j) \tag{13}$$

where s_{i-1} represents the previous hidden state of the decoder, and h_j denotes the hidden state of the encoder at position j. The function $a(\cdot)$ determines the relevance between these states and can be implemented as a dot product, an additive scoring, or a multiplicative scoring function.

The context vector c_i for each decoding step is calculated as a weighted sum of the hidden states of the encoder using the following formula:

$$c_i = \sum_{j=1}^{L} a_{ij} h_j \tag{14}$$

Global attention addresses the limitations of fixed-length context vectors by allowing the model to focus on relevant input segments at each decoding step, improving performance in long and complex texts [27], [35]. Unlike self-attention, which models intrasentence dependencies, global attention dynamically shifts focus across the entire input, making it well-suited for long-text sentiment analysis.

2.5.3 Fully Connected Layer

The context vector c, which represents the extracted features, is passed through one or more dense layers. These layers apply a linear transformation followed by a non-linear activation function to map the features into a suitable space for classification. ReLU (Rectified Linear Unit) activation is used in the hidden layers to introduce non-linearity, improving the model's ability to capture relationships between features. ReLU is defined in Eq. (15):

$$\mathrm{ReLU}(x) = \max(0, x) \tag{15}$$

For the final output, the dense layer uses a softmax activation function, which converts logits into probabilities for multiclass classification. The output probabilities are computed as:

$$\mathrm{y}_{\mathrm{pred}} = \mathrm{softmax}(W_{fc} + b_{fc}) \tag{16}$$

where W_{fc} and b_{fc} are the weight matrix and bias of the dense layer, the softmax function ensures the outputs represent probabilities, which are normalized to sum to one, making the output interpretable as class probabilities.

2.6 Training and Evaluation

In this study, the evaluation will be conducted using an evaluation matrix after the model has been built. The evaluation matrix is used to measure the classification results and determine the effectiveness of the built model. Evaluation metrics include Accuracy, Precision, Recall, and F1-Score. The evaluation metrics used in this research include the confusion matrix, accuracy, precision, recall, and F1-score. These metrics are explained in detail below.

2.6.1 Confusion Matrix

The confusion matrix represents the following: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP refers to an observation correctly identified as belonging to the positive class, while TN represents an observation accurately classified as part of the negative class. Meanwhile, FP occurs when a negative class instance is incorrectly classified as positive, and FN occurs when a positive class instance is mistakenly identified as negative.

Accuracy is the ratio of correct predictions (both positive and negative) to the total number of predictions. The formula for accuracy is given below:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{17}$$

Precision is the ratio of true positive predictions to the total predicted positive cases. It is calculated using the following equation:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{18}$$

Recall is the ratio of true positive predictions to the total of actual positive cases. The formula for recall is shown in Eq. (19):

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{19}$$

The F1 score is the harmonic mean of precision and recall and provides a single measure of model performance. The formula for the F1 Score is presented below:

$$\mathrm{F1} - \mathrm{Score} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \tag{20}$$

3 Results

This section presents the performance evaluation of the proposed sentiment classification model in various scenarios, including hyperparameter tuning, word embedding selection, and attention placement within hybrid deep learning architectures. The model is tested on two datasets: IMDB movie reviews and US Airline Twitter.

3.1 Presentation of Data

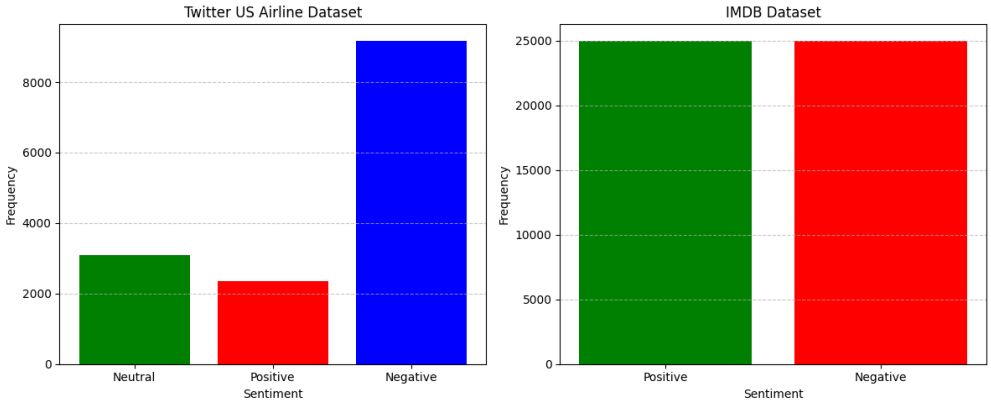

Figure 6: Sentiment distribution on US Airline and IMDb Datasets.

Figure 6 illustrates their sentiment distribution. This study employs two publicly available datasets, IMDb Movie Reviews and Twitter US Airline Sentiment. The IMDb dataset comprises 50,000 balanced movie reviews, evenly divided between positive and negative sentiments. In contrast, the Twitter US Airline dataset comprises 14,640 tweets categorized into positive (16.14%), neutral (21.17%), and negative (62.69%) sentiments, resulting in a highly imbalanced dataset. These tweets originate from customer feedback on major U.S. airlines, including Delta, US Airways, Virgin America, Southwest, and United. The IMDb dataset provides structured and well-organized text suitable for evaluating sentiment models on longer narratives. Meanwhile, the Twitter dataset features short, informal, and unbalanced data, which pose challenges in correctly identifying underrepresented sentiments.

3.2 Hyperparameter

Hyperparameter tuning is performed to determine the optimal hyperparameter values, ensuring that the model achieves its best performance. In this research, hyperparameter experiments were conducted using the same configuration on both the IMDB and Twitter US Airline datasets, with RoBERTa large as the word embedding. Hyperparameter experiments focused on testing the learning rate (l) and the number of hidden units (h). The learning rates tested were $l = \{0.001, 0.0001, 0.00001\}$, and the hidden units tested were

$h = \{128, 256, 512\}$. A detailed description of the experimental scenarios is presented in Table 1.

Table 1: Hyperparameters setting

Hyperparameters	**Values**
Word Embedding	RoBERTa embeddings (pretrained)
Recurrent Neural Networks (RNNs)	CNN-LSTM, CNN-BiLSTM
Optimization method	AdamW
Loss function (L)	CrossEntropyLoss
Epochs	50
Dropout (d)	0.3
Learning rates (l)	$\{0.001, 0.0001, 0.00001\}$
Hidden units (h) of RNNs	$\{128, 256, 512\}$
Training data	90%
Testing data	10%

CrossEntropyLoss is utilized as the loss function, as it is well-suited for multi-class classification, ensuring the model minimizes discrepancies between predicted probabilities and actual labels. The model is trained for 50 epochs, providing sufficient learning time while mitigating the risk of overfitting. To further enhance generalization, a dropout rate of 0.3 is applied, randomly deactivating neurons during training. Additionally, L2 regularization ($\lambda = 0.001$) is incorporated to prevent excessive weight magnitudes, maintaining model stability and adaptability to new data. This approach ensures the model generalizes effectively rather than memorizing training patterns.

Table 2: Hyperparameter experiments result

Models	**LR**	**HU**	**IMDB Dataset**				**Twitter US Airline**			
			Prec	**Rec**	**F1**	**Acc**	**Prec**	**Rec**	**F1**	**Acc**
CNN-BiLSTM	**0.001**	256	**0.9105**	**0.9104**	**0.9104**	**0.9105**	0.7792	0.7855	0.7815	0.7855
CNN-BiLSTM	**0.0001**	256	0.9033	0.9032	0.9032	0.9033	**0.7999**	**0.8046**	**0.8015**	**0.8046**
CNN-LSTM	**0.001**	256	0.9089	0.9088	0.9088	0.9089	**0.8043**	**0.8087**	**0.8044**	**0.8087**
CNN-LSTM	**0.001**	512	**0.9131**	**0.9128**	**0.9128**	**0.9131**	0.7863	0.7937	0.7839	0.7937

In the IMDB dataset, CNN-BiLSTM performed best with a learning rate of 0.001 and 256 hidden units. In comparison, CNN-LSTM required 512 hidden units for optimal performance, indicating that CNN-BiLSTM requires a larger capacity to capture sequential patterns. The results of learning rate experiments for CNN-BiLSTM and CNN-LSTM models are presented in Table 2. In the Twitter dataset for the US Airline, CNN-BiLSTM achieves a learning rate of 0.0001 and 256 hidden units, while CNN-LSTM, with a learning rate of 0.001 and 256 hidden units, shows comparable results but requires careful tuning. In general, CNN-BiLSTM converges faster with a higher learning rate on balanced data, whereas CNN-LSTM requires a lower learning rate and more hidden units for stability. CNN BiLSTM excels in contextual extraction but is computationally expensive, whereas CNNLSTM is more efficient, but demands careful hyperparameter tuning.

3.3 Experiment Result

To evaluate the performance of the hybrid model with attention mechanisms, experiments were conducted using self-attention and global attention at various placements in CNN-BiLSTM and CNN-LSTM. The models utilized the best hyperparameter settings from previous experiments, optimized for each dataset. The evaluation was carried out in two scenarios, comparing a conventional hybrid deep learning model without attention to an enhanced model that incorporates attention mechanisms at various positions.

3.3.1 IMDB Long Text Results

This research focuses primarily on datasets with varying text lengths. The first dataset used is IMDB Movie reviews. This section presents analyses of self-attention and global attention applied to CNN-BiLSTM and CNN-LSTM models. Detailed experimental results are presented in Table 3 and Table 4.

Table 3: Self-attention performance on IMDB Dataset

Models	Position	F1-score	Precision	Recall	Accuracy
CNN-BiLSTM	Without attention	91.04%	91.04%	91.05%	91.05%
	Before BiLSTM	92.13%	92.13%	92.13%	92.13%
	After BiLSTM	92.25%	92.25%	92.25%	92.25%
	Before and After BiLSTM	**93.77%**	**93.77%**	**93.77%**	**93.77%**
CNN-LSTM	Without attention	91.28%	91.28%	91.31%	91.31%
	Before LSTM	92.32%	92.33%	92.32%	92.32%
	After LSTM	92.48%	92.48%	92.48%	92.48%
	Before and After LSTM	**93.53%**	**93.53%**	**93.53%**	**93.53%**

Table 4: Global attention performance on IMDB Dataset

Models	Position	F1-score	Precision	Recall	Accuracy
CNN-BiLSTM	Without attention	91.04%	91.04%	91.05%	91.05%
	Before BiLSTM	**93.23%**	**93.23%**	**93.23%**	**93.23%**
	After BiLSTM	92.86%	92.86%	92.86%	92.86%
	Before and After BiLSTM	93.10%	93.10%	93.10%	93.10%
CNN-LSTM	Without attention	91.28%	91.28%	91.31%	91.31%
	Before LSTM	**93.18%**	**93.18%**	**93.18%**	**93.18%**
	After LSTM	92.55%	92.55%	92.55%	92.55%
	Before and After LSTM	92.18%	92.18%	92.18%	92.18%

Self-attention and global attention enhance the performance of CNN-BiLSTM and CNN-LSTM in IMDB sentiment classification. The best results were achieved when self-attention was applied both before and after recurrent layers, with CNN-BiLSTM reaching an F1 score and accuracy of 93.77% (+2.72%), and CNN-LSTM improving by +2.22%. This setup captures long-range dependencies more effectively, albeit at a higher computational cost. Global attention performed best when applied before the recurrent layers (BiLSTM:

93.23%, LSTM: 93.18%), indicating that early integration of global context facilitates feature selection. However, applying it after or at both stages slightly reduced performance. Overall,self-attention remains the more effective choice for long-text sentiment tasks due to its superior handling of long-range dependencies.

3.4 Twitter US Airline Short Text Results

Performance results are presented in Table 5 and Table 6. Experiments on self-attention and global attention demonstrate their influence on CNN-BiLSTM and CNN-LSTM models when handling short, imbalanced texts. Self-attention yielded the best results when applied both before and after sequential layers, achieving 82.38% accuracy (+1.92%) for CNN-BiLSTM and 81.28% (+1.09%) for CNN LSTM. However, applying it at only one stage (e.g., after BiLSTM) led to a performance drop (CNN-BiLSTM: 80.19%), potentially due to the amplification of dominant class patterns, which hinders the recognition of minority classes. Moreover, self-attention substantially increases computational cost, making it less suitable for short-text datasets unless optimized.

Table 5: Self-attention performance on Twitter US Airline Dataset

Models	Position	F1-score	Precision	Recall	Accuracy
CNN-BiLSTM	Without attention	80.15%	79.99%	80.46%	80.46%
	Before BiLSTM	80.14%	79.92%	80.60%	80.60%
	After BiLSTM	79.59%	79.40%	80.19%	80.19%
	Before and After BiLSTM	**82.70%**	**83.35%**	**82.38%**	**82.38%**
CNN-LSTM	Without attention	80.44%	80.43%	80.87%	80.87%
	Before LSTM	79.34%	79.15%	80.05%	80.05%
	After LSTM	79.53%	79.49%	80.46%	80.46%
	Before and After LSTM	**80.60%**	**81.08%**	**81.28%**	**81.28%**

Table 6: Global attention performance on Twitter US Airline Dataset

Models	Position	F1-score	Precision	Recall	Accuracy
CNN-BiLSTM	Without attention	80.15%	79.99%	80.46%	80.46%
	Before BiLSTM	79.54%	79.79%	79.64%	79.64%
	After BiLSTM	**81.52%**	**81.65%**	**81.69%**	**81.69%**
	Before and After BiLSTM	79.88%	79.88%	81.01%	81.01%
CNN-LSTM	Without attention	79.91%	79.76%	80.19%	80.19%
	Before LSTM	79.26%	79.15%	79.78%	80.73%
	After LSTM	79.46%	79.61%	80.34%	80.34%
	Before and After LSTM	**79.95%**	**79.90%**	**80.87%**	**80.87%**

For global attention, the best performance was observed when applied after BiLSTM in CNN-BiLSTM (81.69%) and before and after LSTM in CNN-LSTM (80.87%). Unlike in long-text datasets like IMDB, global attention in the Twitter US Airline dataset performed better when placed after the sequential layers, refining learned patterns rather than influencing early feature extraction. Although less computationally intensive than self-attention, the

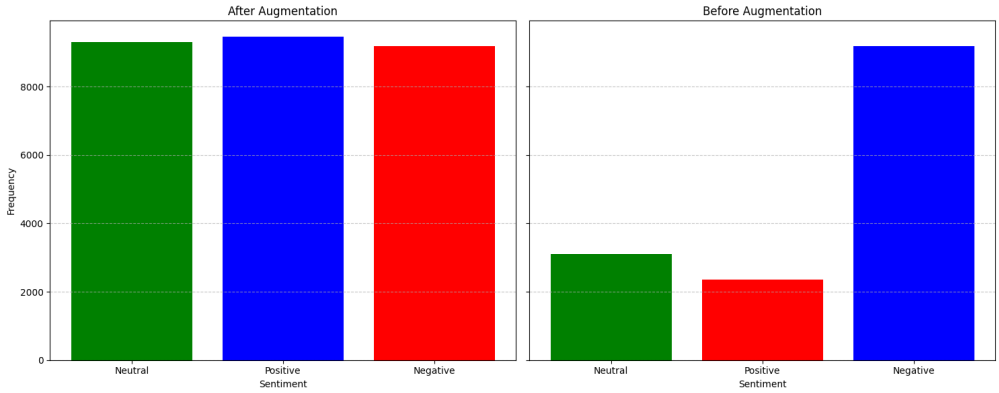

Figure 7: Comparison of data distribution before and after augmentation on the Twitter US Airline Dataset.

performance gains from global attention remain modest, and the risk of overfitting persists due to class imbalance.

3.4.1 Impact of Data Augmentation

Data augmentation can improve performance, particularly for the unbalanced Twitter US Airline dataset. Synonym replacement was used to generate additional samples for under-represented classes (neutral and positive), effectively balancing the data and enhancing the model's ability to capture a broader range of sentiment patterns. As illustrated in Figure 7, data augmentation successfully reduced the dominance of the negative class. As a result, CNN-BiLSTM with self-attention (after BiLSTM) achieved a substantial F1 Score increase from 79.59% to 86.67% (+7.08%). In comparison, CNN-LSTM with global attention (after LSTM) improved from 79.46% to 84.37% (+4.91%), as shown in Table 7.

Table 7: Comparison result before and after data augmentation

Model	**Before Aug.**	**After Aug.**
CNN-BiLSTM + Self-Att. (Post)	79.59%	86.67% (+7.08)
CNN-LSTM + Global Att. (Post)	79.46%	84.37% (+4.91)

4 Discussion

This study demonstrates that the addition of an attention mechanism effectively enhances the performance of hybrid CNN-RNN models for sentiment classification. However, we acknowledge several limitations. This research utilized only two datasets, IMDB and Twitter US Airline, both of which are English-based and US-centric. Limits the generalizability of our findings to other domains, languages, and cultural contexts. Furthermore, class imbalance, particularly in the Twitter dataset, remained a significant challenge despite the application of data augmentation.

We also found that model performance was highly sensitive to variations in hyperparameters, where minor adjustments could significantly influence conclusions about the optimal placement of the attention mechanism. Additionally, the computational cost of self-attention was notable, especially for long-text datasets like IMDB, which could constrain its use in real-time applications. Our data augmentation strategy was also limited to synonym replacement, whereas more sophisticated techniques, such as back-translation, could yield better results. Despite these limitations, our results demonstrate competitive performance compared to previous studies. In the IMDB dataset, our model achieved an F1 score of 93.77%, outperforming the Self-Attention LSTM model of Kardakis et al. [7], which reported an accuracy of 89.71%. A similar trend was observed in the US Airline Twitter dataset, where our F1 score of 86.67% surpassed that of the hybrid RoBERTa-BiLSTM model proposed by [6], which achieved a score of 80.73%. This comparison suggests that strategically placing attention enhances the performance of texts of varying lengths. Therefore, the main contribution of this study is not just achieving the highest score, but rather presenting a systematic analysis of the impact of the position of the attention mechanism, an aspect that comparative studies have not explored in depth.

5 Conclusion

This study explored sentiment classification using CNN-BiLSTM and CNN-LSTM architectures enhanced with self-attention and global attention mechanisms, leveraging RoBERTa embeddings in long-text (IMDb) and short-text (Twitter US Airline) datasets. The results show that self-attention consistently outperforms global attention, especially when applied before and after sequential layers. The CNN-BiLSTM best configuration with dual self-attention achieved 93.77% F1-score on IMDb (+2.72%) and 82.70% F1-score on Twitter (+2.24%).

However, for short, noisy, and imbalanced texts like Twitter, combining self-attention with global attention improves robustness, particularly after data augmentation; however, the gains remain more limited compared to long-form texts due to the fewer contextual dependencies that can be exploited. Post-augmentation, the F1-score increased to 86.67% (+4.29%). These findings emphasize the importance of strategic attention placement and dataset-specific tuning, including augmentation, to maximize model effectiveness. Future work may extend these insights to real-time applications and explore advanced attention designs, such as hierarchical or multi-head attention, to further generalize across domains.

References

[1] J. Y. M. Nip and B. Berthelier, "Social media sentiment analysis," *Encyclopedia*, 2024.

[2] H. Deng, D. Ergu, F. Liu, Y. Cai, and B. Ma, "Text sentiment analysis of fusion model based on attention mechanism," *Procedia Computer Science*, 2022.

[3] N. C. Dang, M. N. Moreno-García, and F. D. la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, 2020.

[4] M. Kamyab, G. Liu, and M. Adjeisah, "Attention-based cnn and bi-lstm model based on tf-idf and glove word embedding for sentiment analysis," *Applied Sciences*, 2021.

[5] L. Shang and Others, "Sentiment analysis of film reviews based on cnn-blstm attention," in *Journal of Physics: Conference Series*, 2020.

[6] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "Roberta-bilstm: A context-aware hybrid model for sentiment analysis," *arXiv preprint arXiv:2406.00367*, 2024.

[7] S. Kardakis, I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis, "Examining attention mechanisms in deep learning models for sentiment analysis," *Applied Sciences*, 2021.

[8] J. Deng and Y. Liu, "Research on sentiment analysis of online public opinion based on roberta–bilstm–attention model," *Applied Sciences*, 2025.

[9] M. Wnkhade, C. Sekhar, and A. Abraham, "Mapa bilstm-bert: Multi-aspects position aware attention for aspect level sentiment analysis," *The Journal of Supercomputing*, 2023.

[10] M. Wankhade, C. S. R. Annavarapu, and A. Abraham, "Cbmafm: Cnn-bilstm multi-attention fusion mechanism for sentiment classification," *Multimedia Tools and Applications*, 2024.

[11] N. A. Semary, W. Ahmed, K. Amin, P. Pławiak, and M. Hammad, "Improving sentiment classification using a roberta-based hybrid model," *Frontiers in Human Neuroscience*, vol. 17, 2023.

[12] K. K. Mohbey, G. Meena, S. Kumar, and K. Lokesh, "A cnn-lstm-based hybrid deep learning approach for sentiment analysis on monkeypox tweets," *New Generation Computing*, 2023.

[13] H. Li, Y. Ma, Z. Ma, and H. Zhu, "Weibo text sentiment analysis based on bert and deep learning," *Applied Sciences*, 2021.

[14] H. Xia, C. Ding, and Y. Liu, "Sentiment analysis model based on self-attention and character-level embedding," *IEEE Access*, 2020.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2023.

[16] W. Wei, Z. Wang, X. Mao, G. Zhou, P. Zhou, and S. Jiang, "Position-aware self-attention based neural sequence labeling," *Pattern Recognition*, 2021.

[17] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based lstm with position context for aspect-level sentiment classification," *IEEE Access*, 2019.

[18] Q. Sifak and E. B. Setiawan, "Hate speech detection using cnn and bigru with attention mechanism on twitter," in *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 2023.

[19] L. Narayanan, "Imdb dataset of 50k movie reviews." https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews, 2020. Accessed: 2025-01-06.

[20] Crowdflower, "Twitter airline sentiment dataset." https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment, 2015.

[21] M. Mahamud, Z. Lee, and I. Samsten, "Distributional data augmentation methods for low resource language," *arXiv preprint arXiv:2309.04862*, 2023.

[22] K. L. Tan, C. P. Lee, and K. M. Lim, "Roberta-gru: A hybrid deep learning model for enhanced sentiment analysis," *Applied Sciences*, 2023.

[23] M. A. Jahin, M. S. H. Shovon, M. F. Mridha, M. R. Islam, and Y. Watanobe, "A hybrid transformer and attention based recurrent neural network for robust and interpretable sentiment analysis of tweets," *Scientific Reports*, 2024.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] R. Gangundi and R. Sridhar, "Rbca-ets: Enhancing extractive text summarization with contextual embedding and word-level attention," *International Journal of Information Technology*, 2024.

[26] Z. Liu, D. Zhang, G. Luo, *et al.*, "A new method of emotional analysis based on cnn–bilstm hybrid neural network," *Cluster Computing*, 2020.

[27] B. Paneru, B. Thapa, and B. Paneru, "Sentiment analysis of movie reviews: A flask application using cnn with roberta embeddings," *Systems and Soft Computing*, vol. 7, p. 200192, 2025.

[28] S. Soni, S. S. Chouhan, and S. S. Rathore, "Textconvonet: A convolutional neural network based architecture for text classification," *Applied Intelligence*, 2022.

[29] D. Zhang, L. Tian, M. Hong, H. Fei, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, 2018.

[30] W. Yue and L. Li, "Sentiment analysis using word2vec-cnn-bilstm classification," in *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2020.

[31] Y.-H. Hsieh and X.-P. Zeng, "Sentiment analysis: An ernie-bilstm approach to bullet screen comments," *Sensors*, 2022.

[32] A. Pimpalkar and J. R. R. R, "Mbilstmglove: Embedding glove knowledge into the corpus using multi-layer bilstm deep learning model for social media sentiment analysis," *Expert Systems with Applications*, 2022.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] 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*, 2022.

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[36] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *CoRR*, 2024.

[37] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.