Multi-label Classification for Sentiment Analysis Using CBGA Hybrid Deep Learning Model

Doha Taha Nor El-Deen, Rania Salah El-Sayed, Ali Mohamed Hussein, and Mervat S. Zaki

***Abstract*—In recent years, many real-time applications extensively utilize text classification problems. It is vital to make use of text classification methods by developing new text mining and applications of natural language processing (NLP). Among the most well-known uses of text classification are intent detection, language recognition, topic labelling, and sentiment analysis. An in-depth examination of deep learning techniques is necessary because of the exponential rise in the quantity of complicated documents. Any deep learning algorithm's capacity to comprehend the nonlinear relationships between complicated models within data is what determines its success. Therefore, A researcher's formidable task is to develop appropriate techniques, structures, and models to classify texts. This study suggests a blended deep-learning framework depending on the attention mechanism and is well-analyzed for text categorization. The suggested hybrid approach entails a Convolutional Bidirectional Gated Recurrent Unit (Bi-GRU) featuring an attention mechanism and output (CBGA) model. In this model, the attention mechanism is put after the Bi-GRU and then the construction output SoftMax layer. The model was implemented and operated on more than one dataset. According to the comparative analysis, the suggested CBGA model performs better than the other one that was chosen using the conventional approach. We obtained a high classification accuracy on some of the datasets, and these results are the optimum that has been achieved so far.**

***Index Terms*— deep-learning, convolutional neural networks, attention mechanism, and Bidirectional gated recurrent.**

I. Introduction

THE text classification is represented in NLP analysis. In the domains of information retrieval, natural language processing, and web mining, text classification is essential for managing massive enormous volumes of text documents [1]. Efficient utilization of information derived from textual content is paramount, particularly in the field of categorization of texts, where the allocation of predefined topics to natural language documents is seamlessly achieved. Text classification leverages diverse machine learning techniques [2], including but not limited to neural networks, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Bayesian methods [3],[4],[5],[6]. In figure 1 demonstrates machine learning techniques. These methodologies, encapsulated within the broader scope of machine-learning, contribute significantly to the automated classification for textual data.

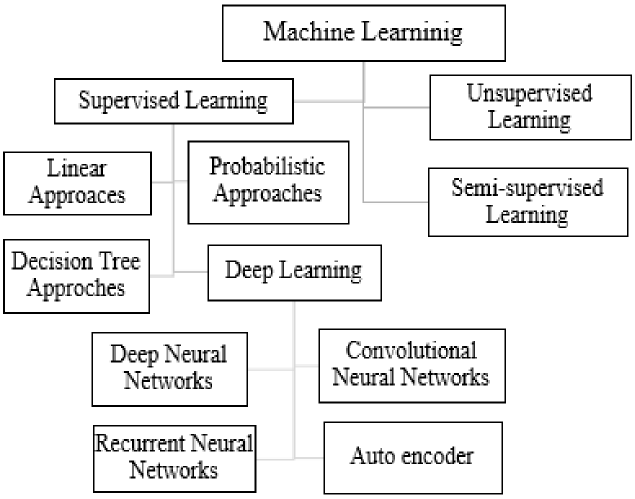

Fig.1: Machine learning approaches.

During the previous several years, Deep learning methods have advanced significantly in terms of text classification. Neural networks, both recurrent (RNN) and convolutional, (CNN) can attain good performance when implemented for text classification, sentiment analysis, image recognition, topic type labeling, etc.[7],[8]. Deep learning has become a prominent force, making significant strides across diverse industries. Among its key components are Artificial Neural Networks (ANNs), which emulate the intricate workings of the human brain by constructing a hierarchy of representations through complex structures and multi-layer models [9]. The applications of deep learning extend seamlessly into the field of Natural Language Processing (NLP), where the automatic analysis and representation of human languages prove immensely advantageous. Lately, Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have garnered substantial attention in the domain of NLP. Their widespread adoption is fueled by their exceptional

Manuscript received April 18, 2023; revised December 17, 2023

Doha Taha Nor El-Deen is an assistant lecturer at the Centre of Basic Science, Misr University for Science and Technology, Giza, Egypt. (e-mail: doha.taha.1911@gmail.com)

Rania Salah El-Sayed is an Associate Professor at the Department of Mathematics, Faculty of Science (girls branch), Al-Azhar University, Cairo, Egypt. (e-mail: rania5salah@azhar.edu.eg)

Ali Mohamed Hussein is a professor at the Department of Mathematics at the Centre of Basic Science, Misr University for Science and Technology, Giza, Egypt. (e-mail: ali.salem@must.edu.eg)

Mervat S. Zaki is a professor at the Department of Mathematics, Faculty of Science (girls branch), Al-Azhar University, Cairo, Egypt. (e-mail: mervatzaki.1959@azhar.edu.eg)

performance in assignments for example, sentiment analysis, text categorization and summarization [10]. Researchers are drawn to these neural network architectures as they exhibit remarkable capabilities in understanding and processing the intricacies of language. The fusion of deep learning and NLP holds promise for revolutionizing how we interact with and derive insights from textual data in various applications and industries. The improved architecture of RNN is Long Short-Term Memory (LSTM), wherein a gate mechanism consisting of input gate, a forgotten gate, and an output gate is applied [11]. An RNN type called LSTM networks uses special units in addition to regular units. The memory cell inside LSTM units can save information for extended lengths of time in memory. These memory cells allow them to learn longer-term dependencies. Along with effectively resolving the problem of vanishing gradient, the gate mechanism has also resolved the long-term data conservation issue. In text classification, the superior ability of LSTM to extract textual information in a versatile manner plays an important role. In the last few years, the usefulness of the LSTM has been greatly explored, and Researchers are constantly refitting or altering the LSTM to increase its accuracy even more.

In Section II, we delve into the landscape of similarity work, exploring the advancements and foundations that pave the way for our research. Building upon this contextual understanding, Section III unfolds the intricacies of techniques of deep learning tailored specifically for classification of text. The core of our contribution takes center stage in Section IV, where we meticulously unveil the architecture of the main blocks of building that constitute our suggested model. Section V stands as the arena for results and discussion, where the empirical outcomes and their implications engage in a compelling dialogue. Finally, the journey concludes in Section VI, where we distill insights, summarize findings, and chart the course for future exploration.

II. RELATED WORK

Most recent work involves deep learning to classify text efficiently we will mention it as below.

In the realm of text classification, the organization of information plays a pivotal role in comprehending the landscape of advancements. Delving into related work, A. Joulin et al. [12] introduced a method leveraging product quantization for the storage of word embeddings. Their approach involved the strategic application of discriminative pruning, a technique designed to retain only crucial features within the trained model. Remarkably, this method not only demonstrated resilience in preserving the integrity of information but also exhibited a noteworthy balance between accuracy and memory usage.

M. Iyyer et. al. [13] presented the Deep Average Networks (DAN) that is an illustration of feedforward neural networks which utilized for text representation. The DANs, unlike more complex composition functions, perform effectively on data that have high syntactic variance and explicitly model semantic and obtain high accuracy.

In the ever-evolving landscape of natural language processing, K. S. Tai et al. [14] pioneered a significant advancement with their introduction of a novel LSTM variant. This innovation marked a departure from the conventional sequential word-based RNN models, as they embraced a more intricate approach—tree-structured network topologies. This departure yielded remarkable results, with their Tree-LSTMs showcasing superior performance when compared to robust LSTM baselines across diverse tasks. The prowess of Tree-LSTMs particularly shone in tasks involving sentiment classification, where the model exhibited a nuanced understanding of context and emotion. Additionally, the network demonstrated its capabilities in predicting semantic relationships between two sentences, showcasing its versatility in capturing intricate linguistic nuances. This breakthrough underscores the potential of exploring alternative network architectures in the realm of deep learning, opening avenues for more sophisticated and context-aware language models. The work by K. S. Tai et al. [14] not only contributes to the ever-growing arsenal of NLP techniques but also prompts further exploration into the realm of tree-structured neural networks and their potential applications in understanding and processing natural language.

X. Zhu et. al. [15] presented S-LSTM is extension of LSTM to tree structures. S-LSTM wires memory blocks in a partial-order tree structure instead of in a full-order the same as in a chain-structured LSTM. It outperforms a modern iterative model by replacing its configuration layers with S-LSTM memory blocks, its performance on the test set of Stanford Sentiment Treebankat: the sentence level (ROOTs) is 48.9% and the phrase level is 81.9% and show that specific structures is beneficial for achieving better performance than without considering structures.

Y. Kim [16] conducted a series of insightful experiments that highlighted the effectiveness of Convolutional Neural Networks (CNN) when integrated with word2vec in the realm of NLP (Natural Language Processing). Surprisingly, even a straightforward CNN architecture featuring just one layer of convolution displayed remarkable performance, requiring minimal hyperparameter tuning. This underscores the robustness and efficiency of this approach in handling complex language structures. Kim's findings contribute significantly to the ongoing exploration of deep learning techniques for NLP, offering a valuable perspective on the potential simplicity and effectiveness that can be achieved in designing neural networks for language-related functions.

Zhang et. al. [17] presented an effective method to use ConvNets (character-level convolutional networks) for classification of text. We compared many deep learning and traditional models using many of large-scale datasets to attain state-of-the-art or competitive outcomes.

J. Prusa and T. Khoshgoftaar's innovative approach, as outlined in their work [18], tackles the challenge of optimizing memory usage and training time in convolutional neural network (CNNs) when dealing with character-level text representations. By implementing a text encryption strategy, they significantly enhance the efficiency of CNNs in processing and learning from textual data. This encryption technique not only streamlines the training process but also minimizes the memory footprint required for effective model performance. Their research stands as a valuable contribution to advancing the capabilities of CNNs in handling character-level text representations with increased speed and resource optimization.

Z. Yang et. al.[19] introduced HAN (hierarchical attention network) for classification of documents that is structured hierarchically, that is modelled after the document hierarchy. the dual attention mechanisms present within the Hierarchical Attention Network (HAN), it applied at the sentence-level and words. This model is effective in picking out important words and sentences which visualize these attention layers illustrate. Test a model in data sets such as Yelp 2013 and IMDB, the accuracy is 71% and 49.4% respectively.

X. Zhou et. al. [20] have advanced significantly. In the field of natural language processing with their attention-based bilingual representation learning model. This innovative approach delves into the intricacies of distributed semantics within documents, bridging the gap between target languages and source. The model employs a hierarchical attention mechanism, elevating the capabilities of the bilingual LSTM network. By focusing on the nuanced relationships within and between languages, Zhou et al. have laid the foundation for more effective and nuanced language understanding, promising advancements in cross-language applications and document analysis. The authors' model achieved good results on a benchmark dataset. LSTM+HA combines both word-level and sentence-level attention; the average accuracy was achieved 82.4%.

T. Shen et. al. [21] introduced a new mechanism of interest wherein attention between elements of the input sequence is multidimensional and directional. A lightweight neural network, "Directional Self-Attention Network (DiSAN)", is proposed for learning sentence embedding, based only on the suggested interest minus any RNN/CNN architecture.

Y. Liu et. al. [22] presented a sentence encoding-based model for recognizing text in Natural Language Inference. In the first stage, they generated sentence representation using average pooling across word-level bidirectional LSTM (bi-LSTM). Then, to get a more realistic depiction, they employed the attention mechanism on the same sentence rather than average pooling. 85% accuracy has been reached.

H. Peng et. al. [23] presented a deep learning model to implement largescale hierarchical classification of text based on a CNN graph to convert texts into a word graph first, and then skew the word graph using graph convolution techniques.

L. Yao et. al. [24] introduced a new method of classification text termed using graph convolutional networks (GCN). A Text Graph Convolutional Network (Text GCN) is learned for the corpus after they construct a single text graph for the corpus based on a heterogeneous document word relations and word co-occurrence. Text GCN accomplishes 86.34% for an accuracy of the 20NG database and accuracy of 68.36% for Ohsumed database, which is higher than some baseline models.

Felix Wu et. al. [25]Graph Convolutional Networks (GCNs) which have been considered as ways to learn graph representations (GCNs) but refer to the simplified linear model as Simple Graph Convolution (SGC). By eliminating nonlinearity and collapsing weight matrices between successive layers, it lessens overcomplexity. We experimentally show that the final linear model displays similar or even superior performance to that of GCNs in a variety of tasks while being computationally more efficient and suitable for significantly fewer parameters.

A. Vaswani et. al. [26] introduced the Transformer, the first fully attention-dependent sequential transformation model, eliminating redundancy and convolution entirely, substituting multi-headed self-attention for the recurrent layers that are often utilised in encoder-decoder systems. When using the standard WMT 2014 English-German dataset, his model (28.4 BLEU) achieved the best results.

J. Chung et. al. [27] presented the more complex units, which implemented a gating mechanism such as GRU and LSTM. The analysis amply illustrated how much better the gating units (GRU-RNN and LSTM-RNN) were in Ubisoft Datasets than the conventional tanh unit.

N. Wang et. al [28] presented a model that uses Bi-GRU with attention mechanism. The combined proposal of Word2vec and GloVe is higher than the basic models. Compared to previous models, the accuracy of the GloVe and Word2vec combined model is superior (accuracy: 71%).

L. Zhang et. al. [29] proposed a hierarchical multi-input and output model based HMIO (bi-directional recurrent neural network), that takes into account both the lexical and semantic information of emotional expression. They apply two independent Bi-GRU layers and attention over output of soft-max activation. The presented model (HMIO) has several features and has Achieve advancements, the accuracy of HMIO is 79% and the accuracy of attention with HMIO is 81% in customer reviews of mobile phones.

J. Wu et al. [30] presented a hierarchical attention network model and added a context layer (CAHAN) to increase accuracy, decrease the amount of irrelevant phrases considered, and identify the sentiment polarity direction. The accuracy of CAHAN on the IMDB dataset is 93.65%. The observations of the proposed model (CAHAN) show this model has higher accuracy and lowers the loss rate and shortens the training period as compared to earlier and current models.

S. Sachin et al. [31] used GRU, LSTM, Bi-GRU and Bi-LSTM techniques to execute sentiment classification and review analysis on Amazon and produced positive outcomes. The LSTM and GRU models are capable of extracting semantic contextual data, however the Bi-GRU model performs better than the other since it obtained a higher score for each performance metric. With an accuracy of 71.19%, the bidirectional gated recurrent units exhibit the highest level of performance.

SK. Prabhakar et al. [32] used two model, which used methods CNN, Bi-LSTM and attention mechanism, the first hybrid model is called (CBAO) model which is consists of convolutional Bidirectional Long Short-Term Memory (Bi-LSTM) with attention mechanism, and the second hybrid model is called (CABO) which is consists of applying Bi-LSTM to a convolutional attention mechanism. The accuracy for the IMDB dataset in CBAO model is 92.72% and the accuracy in CABO model is 90.51%.

Yan Cheng et al. [33] presented a multichannel which blends CNN and bidirectional gated recurrent (BiGRU) with attention mechanism (CNN+AttCNN+AttBiGRU) design. It can extract more rich text features from datasets, so the results of experimental on the Yelp 2015 dataset and the IMDB dataset achieved that the accuracy in first dataset is 91.70% and the accuracy in the second dataset is 92.90%. Table I provides some comparisons of related research on their model accuracy of various datasets.

III. Sentiment Analysis Using Deep Learning Techniques

A. *The Structure of CNN model*

In the first; In computer vision, CNN was utilized, it has been used extensively in NLP and tasks in recent years with good results and has worked better than sequencing-based techniques [16]. Convolutional, pooling, and fully connected layers are what make up CNN primarily. The fully connected layer is comparable to the hidden layer of a conventional feed forward neural network; it is typically connected to the output layer at the end to arrive at the final output. The convolution layer is used in the input data to extract the features from it, and the pooling layer is used to select and filter the features extracted by the convolution layer and filter it [55]. The structure of 1D CNN that is used for classification of text is shown in Figure 2.

TABLE I
COMPARISON ON VARIOUS DATASETS FOR SOME RELATED WORK.

METHOD	DATA BASE	PERFORMANCE ACCURACY
CNN [41]	Movie reviews and IMDB	89.4 %
CNN, LSTM [42]	Stanford sentiment	88.3%
LSTM, CNN [43]	Stock Twits	90.9%
CNN [44]	semEval-2016 Task 1 and 2	87.0%
CNN, RNN [45]	Stanford Twitter sentiment	90.6%
RNN, LSTM, GRU [46]	Review polarity, IMDB	87.2%
CNN, LSTM [47]	Movie reviews	89.02%
CNN-LSTM [48]	Twitter product Reviews	93.85%
CNN, BiGRU, BiLSTM [49]	StackOverflow, ISEAR, EmoInt, DailyDialogs	90.0%
CNN, LSTM [50]	Persian Movie Reviews	92.3%
DNN [51]	Twitter Online Tweets	72.7%
Stacked BiLSTM [52]	US airlines	92.0%
LSTM, CNN [53]	Large Movie Reviews	93.1%
CNN-LSTM [54]	UCL, RUSA-19	84.1%

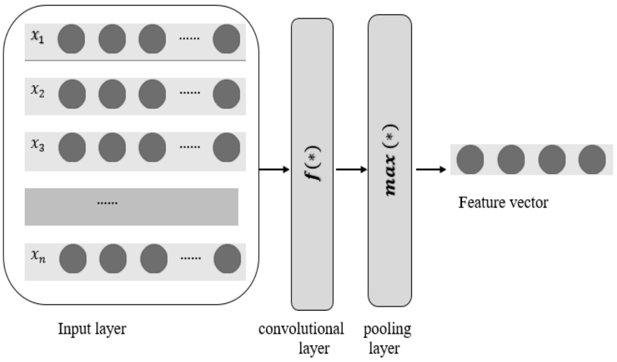

Fig.2: The structure of CNN.

We use a matrix $\in R^{n*l}$ to represent the input sentence in our model, where n denotes the sentence's word count and l the word embedding vector dimension of each individual word. Assume that the convolutional kernel $W^c \in R^{h*l}$ is the convolution kernel's width, its size is equal to the word's dimension, and c is the number of convolution kernels, l the convolution kernel's length, and is the embedding vector. For the input matrix M $\in R^{n*l}$, the feature map vector V=[v_o,v_o, …, v_{n-h}] $\in R^{n-h+1}$ is obtained by Performing a convolution process by frequently applying a convolution kernel W, where each element in the feature vector V is calculated using the formula provided in equation (1).

$$V_i = W \circ M_{i:j+h-1} \qquad (1)$$

Where i =0,1,2,3,…, n-h, ($\circ$) indicates the matrix's point-wise multiplication processing, and $M_{i:j}$ indicates the sub-matrix of the M matrix from i to j rows, for example, the i-th word's word embedding vector matrix to the j-th word. Each of the feature map vectors V that are produced after the convolution operation is transferred to the pooling layer in order to produce possible features and filter the features. The most popular pooling technique is max-pooling, which uses equation (2) to convert a sentence with variable length into one with a fixed length by capturing the most significant feature T following convolution.

$$T = \max_{0 \le i \le n-h} \{V_i\} \qquad (2)$$

B. *GRU*

The excellent RNN variation known as GRU was put forth by Cho et al. [34] to keep memories for long distance dependencies, thus it avoids the vanishing gradient problem similar to LSTM but GRU is a less complex variant compared to LSTM. GRU includes reset gate and update gate. It requires less parameters and has a faster training convergence time than LSTM since it has one fewer gate. The GRU architecture appears in Figure 3.

Using (3) and (4), the reset gate r_t (whose update procedure is like that of the update gate z_t) and update gate z_t, which may calculate the update degree of the activation value in the GRU unit based on the state of the preceding hidden layer and the current input state, are computed.

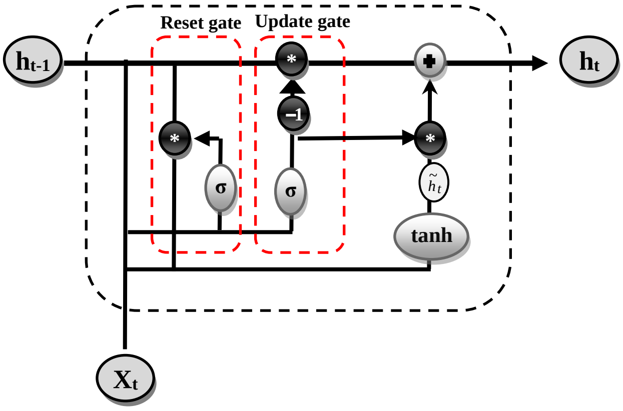

Fig. 3: Architecture of the GRU

$$r_t = \sigma(W_r X_t + U_r h_{t-1}) \quad (3)$$

$$z_t = \sigma(W_z X_t + U_z h_{t-1}) \quad (4)$$

Then, the hidden layer or actual activation at time t of the GRU h_t and $\tilde{h}_t$ is a candidate hidden layer are calculated by using (5) and (6):

$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t \quad (5)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t * h_{t-1})) \quad (6)$$

Where σ represents the logical sigmoid function W_r, W_z , U_r , U_z , U_h , W_h are the weight matrices of GRU and x_t is input at t (time point).

C. Bi-directional GRU

GRU is transmitted words in one direction from front to back, this indicates to disregard the impact of the subsequent words [35]. One kind of GRU that is based on the states of two GRUs is the bidirectional GRU. In other words, it addresses the GRU problem by relying on the dual effects for both backward and forward situations, it resolves the GRU issue, improving the precision of the final output because it allows future and past information to impact the states of current. In Figure 4; the structure of the bidirectional GRU model is shown.

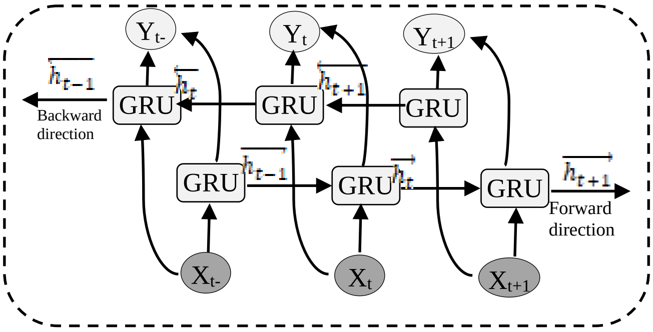

Fig.4: Architecture of the Bi-GRU

The output of the t^th word is concatenated between backward & forward states from Bi-GRU by using equations (7), (8), and (9).

$$\overrightarrow{h_t} = \overrightarrow{GRU}(x_t, \overrightarrow{h_{t-1}}) \quad (7)$$

$$\overleftarrow{h_t} = \overleftarrow{GRU}(x_t, \overleftarrow{h_{t+1}}) \quad (8)$$

$$h_t = \overrightarrow{h_t} \oplus \overrightarrow{h_t} \quad (9)$$

D. The mechanism of attention

Attention mechanisms (ATT) have become integral in various realms of deep learning, finding applications in tasks ranging from NLP to speech recognition. The development of deep learning techniques is intricately linked with the nuanced role of attention mechanisms. As model architectures advance, understanding relies heavily on the attention mechanism. The model allocates emphasis on words based on their similarity, enhancing both learning and generalization capabilities. This intricate dance of attention fosters a more nuanced understanding of context, allowing the model to discern and prioritize information effectively, thereby contributing significantly to the overall efficacy of deep learning systems [36].

In the sentiment classification, Yang et al. [19] achieved optimal results by implementing the attention mechanism at the text chapter level. The attention mechanism, comprising encoding and decoding units, plays a pivotal role. The encoding unit, typically an encoder, undergoes transformations on input data to derive a semantic vector. Meanwhile, the decoding unit, often a decoder, processes the output data through specific transformations. This dual-unit mechanism enhances learning and generalization capabilities, with a focus on words that bear greater resemblance. The nuanced interplay between encoding and decoding ensures an effective and nuanced approach to sentiment analysis in the context of text chapters. The structure of the attention mechanism is illustrated in Figure 5.

Since the relevance of each word varies depending on the sentence, we use equations (10), (11) and (12) to establish an attention mechanism that extracts the semantic information of the significant words in the sentence:

$$u_i = \tanh(b_i + W_i h_i) \quad (10)$$

$$\alpha_i = \frac{\exp(u_i^t u_w)}{\sum_i \exp(u_i^t u_w)} \quad (11)$$

$$h_i = \sum_i h_i \alpha_i \quad (12)$$

Where u_i is the result of a full connection operation of the hidden layer vector. Throughout training, the context vectors h_i and u_w are randomly initialised and updated; b_i and W_i are the bias term of attention and weight matrix calculation respectively, For the i-th word in the sentence, the attention score is denoted by α_i .

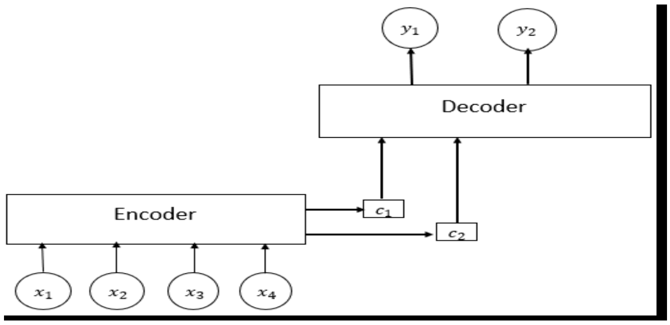

Fig. 5: The architecture of Attention Mechanism

IV. THE OVERALL

In this study, CBGA model is indicated in Figure 6 which contains three main frames data processing, extraction of feature and classification. The main body is composed of 3 CNN model blocks, one bi-directional gated recurrent network (Bi-GRU) model block and attention mechanism layer. The suggested model consists of 5 blocks: word embedding layer, CNN layers, bidirectional GRU layer, attention mechanism layer, classification layer and output SoftMax layer.

The overall model in our paper is dependent on CNN, Bi-GRU with attention mechanism. Initially, the text sentences are tokenized, each word in each sentence gets a digital representation (a number). When a text input is received, the word embedding layer uses a dictionary index to convert the digital representation of the text into a discourse vector. CNN blocks, bi-GRU, and ATT are utilised in the process of extracting features from the text. The embedding layer, which is the initial layer, maps each input word into a vector representation; the second layer, called the convolution layer, is primarily responsible for extracting the local characteristics between words, the word embedding vector's dimension is set to 130×32 in here, the three filter sizes were chosen 3,3,3 with 64,128,256 feature maps each, the stride is set to 1, the padding is set to same (no need to perform zero padding operation). After the convolution process, the sentence's local features can be acquired; The pooling layer, which makes up the third layer, essentially applies max-pooling to the local features that the convolutional layer has produced, It extracts the more significant characteristics between sentences, eliminates certain superfluous and irrelevant information, and creates feature vectors, the second and third layer followed by a dropout are used three times, then the bi-directional GRU channel is used to get sentence context semantic information then the attention mechanism layer used mainly to determine the importance of each word given a specific output, resulting in a matrix of weights that represents how much attention to pay for each word to deduce an output, Richer feature information is provided by the fully connected layer for the next sentence sentiment classification; ReLU is used as the activation function in this layer and a dropout of 20% is applied after each layer (including hidden layers) helps with the regularization to prevent over-fit. Ultimately, with the assistance of Sigmoid output layer, the model's output, which uses the Adam optimizer and the binary-cross entropy loss function, is the result of classification.

V. Experimental Results and Discussion

The setting for the experiment (cloud computing) of our study is as follows: The GPU is tesla K80, the memory size is 78GB, the RAM is 12.68 GB and the development tool uses python 3, Keras version 2.8.0. The datasets examined, as well as the suggested deep learning approach parameter settings and implementation assessment measures.

A. *Experimental dataset*

For experimentation, three datasets were described here.

The first dataset is called IMDB dataset: The dataset (https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews) was created with the intention of binarily classifying movie reviews' sentiment. In this dataset, the core data set contains 50,000 comments, the number of negative and positive reviews is divided equally 25,000 negative reviews and 25,000 positive reviews.

The second dataset is called Yelp2015 reviews: The dataset (https://zenodo.org/record/5259139)is obtained from Yelp dataset Challenge in 2015. There are five rating levels, ranging from 1 to 5.

The third dataset is called Movie Review Data (MR): One sentence per review for each movie in the dataset (http://www.cs.cornell.edu/people/pabo/movie-review-data/). Classification involves binary (positive and negative) categories of reviews.

TABLE II
THE DETAILED HYPER PARAMETER SETTING

hyper-parameter	**Value**
dimension of Word vector	32
Convolution kernel size	(3,3,3)
Batch size	100
Bi-directional GRU hidden layer size	150
Epochs	4
Adam rate	0.001
Dropout rate	0.2

B. *Analysis of Proposed*

Natural language processing heavily relies on word vectors; pre-training with word vectors improves the model's classification accuracy. In the preprocessing, divide each sentence into words by space, lemmatize verbs and adjectives, and then remove the stop words.

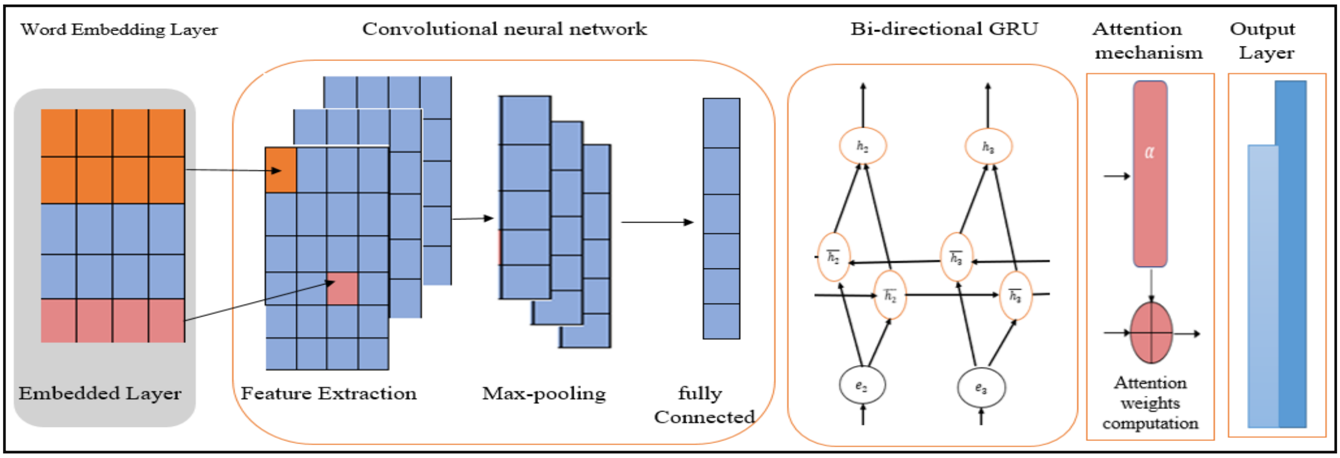

Fig.6: Architecture of the proposed CBGA model for sentiment analysis

In this study, we set the maximum sentence length to 130, we set the word vector dimension to 32. The sentence is zero-padded, if the length is smaller than 130. the sentence is truncated if the length is larger than 130. Based on framework of the Keras deep learning, this paper was written. Adam, the model's optimization function, because of its ability to establish separate adaptive learning rates for various parameters and hasten network convergence. In our study, to find richer information on emotional traits, the number of each kernel of convolution is 64, 128, and 256, and the convolution kernel window size is 3, 3, 3. To avoid overfitting, we employed the regularization mechanism Dropout in these experiments. The details of the hyper-parameter settings for this model appear in Table II.

C. Metrics for Performance Evaluation

In our study, we employ the performance metrics listed below for valuation. When classifier performance accuracy is calculated, it may be stated as:
The precision is means as (13).

$$precision = \frac{TP}{TP + FP} \tag{13}$$

The sensitivity (AKA recall) is means as (14)

$$sensitivity = \frac{TP}{TP + FN} \tag{14}$$

The F score (F_1) means as (15)

$$F_1 = \frac{2 * precision + sensitivity}{precision + sensitivity} \tag{15}$$

The specificity is means as (16)

$$specificity = \frac{TN}{FP + TN} \tag{16}$$

The Matthews Correlation Coefficient (MCC) is means as (17)

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt[2]{(TP + FN) * (TP + FP) * (TN + FP) * (TN + FN)}} \tag{17}$$

The geometric mean $g\text{-}mean$ is expressed as (18)

$$g - mean = \sqrt{\frac{TP * TN}{(TP + FN) * (TN + FP)}} \tag{18}$$

In the final evaluate the accuracy and it is means (19)

$$Accuracy = \frac{TN + TP}{FP + FN + TP + TN} \tag{19}$$

where true positive, false positive, true negative, and false negative are represented, respectively, by the symbols TP, FP, TN, and FN.

D. Experimental comparison

When machine learning has employed deep learning methods like CNN and some conventional techniques for linear approaches like Linear Discriminant Analysis (LDA) and SVM. CNN was proven to be the best of them, we improved it to produce the finest possible outcome. Illustratively, it is in the Table III. The models were implemented and operated on the IMDB, Yelp 2015 and MR datasets. Word representations are learned by a probabilistic model of documents that is derived using LDA, which captures semantic similarities between words. This component is based on probabilistic topic models and does not require labelled data. The reported accuracy result was 67.40% when using IMDB dataset, the reported accuracy result was 68.20% when using Yelp 2015 dataset and the reported accuracy result was 69.11% when using MR dataset. Using the most famous non-deep learning techniques [37].
Authors in [38] used the SVM as classifier for text categorization with sentiment and the reported accuracy result in the order was 79.86%, 81.16% and 71.42%.

TABLE III
COMPARISON OF THE MODELS ADOPTED ON DIFFERENT DATASETS

Model	**IMDB dataset**	**Yelp 2015**	**MR**
LDA [37]	67.40%	68.20%	69.11%
SVM [38]	79.86%	81.16%	71.42%
CNN [16]	88.82%	90.36%	81.31%

The authors in [16] used a simpler form of CNN because a single-channel CNN is a common classification, so embedding each word in the sentence is used to the word vectors, and It is fed into the CNN as its input and subsequently passes via the pooling, convolutional, fully connected, and last SoftMax output layers. The reported accuracy result was 88.82%, 90.36% and 81.31% respectively when using CNN model. They improved the use of CNN in several ways as demonstrated in Table IV.

TABLE IV
COMPARISON OF IMPROVED MODELS ADOPTED ON DIFFERENT DATASETS

Model	**IMDB dataset**	**Yelp 2015**	**MR**
C-LSTM [39]	89.13%	90.80%	81.76%
CNN+GRU [40]	90.01%	90.91%	81.94%
ATT+CNN+BGRUM [35]	90.22%	91.30%	82.83%
MC+CNN+ATTBIGRU [33]	90.82%	91.82%	82.76%
MC+ATTCNN+ATTBIGRU [33]	91.70%	91.90%	83.89%
MultiCNN+BIGRU+ATT [proposed]	**97.75%**	**95.01%**	**87.02%**

The authors in [39] employed long-term dependencies on window feature sequences with LSTM and higher-level word feature sequence extraction with CNN, respectively.

TABLE V
RESULTS OF COMPARISON THE MODELS ON IMDB DATASET WITH DIFFERENT MEASURES.

Methods	Precision (%)	MCC (%)	Sensitivity (%)	Specificity (%)	g-Mean (%)	Accuracy (%)
CBAO [32]	90.21	95.23	94.73	92.68	0.8544	92.72
CABO [32]	95.31	85.71	86.36	90.38	0.809	90.51
CBGA (Proposed)	**97.784**	**97.74**	**96.78**	**97.71**	**0.9773**	**97.75**

Ultimately, the SoftMax layer and the full connection yield the desired outcome. The reported accuracy result in the order was 89.13%, 90.80% and 81.76%.

Authors in [40] presented a novel technique that combines convolutional and gated recurrent networks in a deep neural network (CNN+GRU). This technique can record information about word order and sequencing in brief texts. In order to obtain the classification result, the first approach uses the technique word embedding as the input of the CNN; the second uses the extracted features as the input of the GRU; and the third uses fully connection and SoftMax layer. The reported accuracy results in order were 90.01%, 90.91% and 81.94%.

The authors in [35] introduced a new model depend on a deep neural network combining attention mechanism, convolutional and bi-directional gated recurrent networks (ATT+CNN+BGRU). There is just one channel in this CNN-Bi-GRU model. Using the multi-channel CNN model, the text's different n-gram features are first retrieved. The extracted features are then fed into the Bi-GRU model, which is based on the attention mechanism, and ultimately, max-out neurons are used to provide the classification results. The reported accuracy results were 90.22%, 91.30% and 82.83% respectively.

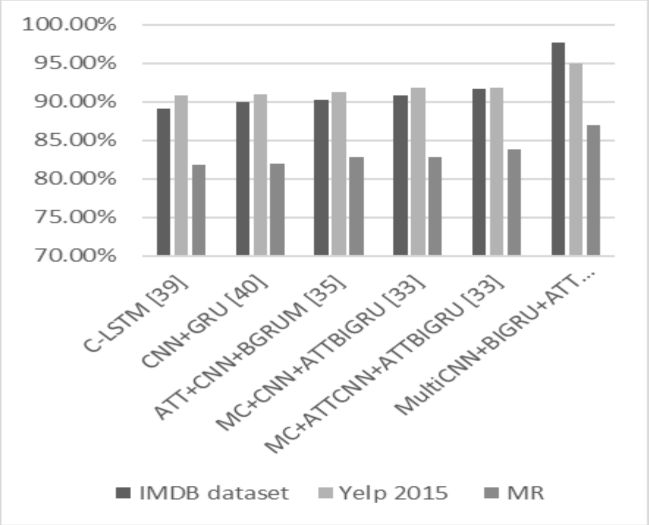

Fig. 7. Comparison between earlier approaches and the proposed approach on different data sets.

The authors in [33] introduced two approaches. The first approach is called MC+CNN+ATTBGRU: which consists of multi-channel CNN (three channels) and one bidirectional GRU channel, then the attention mechanism is added to the bidirectional GRU channel. The reported accuracy results in order were 90.82%, 91.82% and 82.76%.

The second approach is called CNN+AttCNN+AttBiGRU: It is made up of a bidirectional GRU channel and a three-channel CNN. The attention mechanism is added to both the bidirectional GRU channel and the three-channel CNN. Next, the word embedding receives the word vector as input, and the features are extracted via the bidirectional GRU channel and the CNN channel. The reported accuracy result in the order was 91.70%, 91.90% and 83.89% r, and finally the extraction of features is combined to execute the last sentiment classification. Figure 7 represents comparison of some approaches to data sets. Table V presents a comparison analysis of the suggested model and the other models that were implemented.

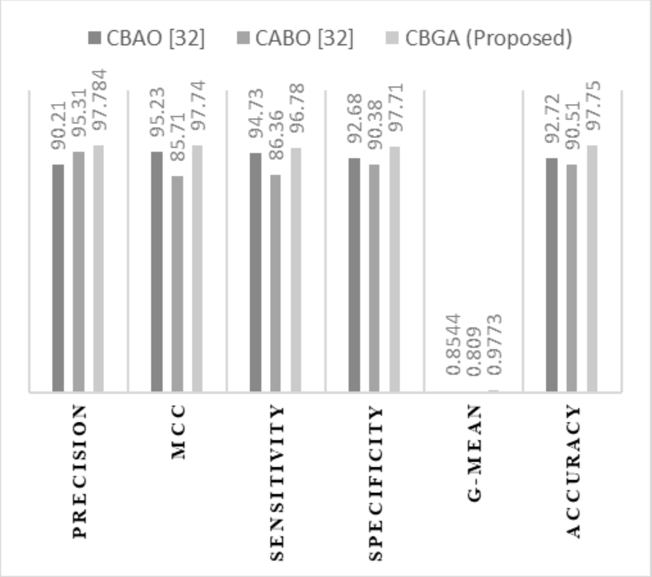

Fig. 8: Performance Measures Analysis for CBAO, CABO and CBGA models

Authors in [46] used the CBAO model based on attention mechanism after bi-directional LSTM. The reported result was (92.72%). Also tested the CABO [46] approach that used attention mechanism before bi-directional LSTM. The reported result was (90.51%). The average accuracy for the proposed model using the CBGA is 97.75% on IMDB dataset. The low error rate of precision in CBAO, CABO and CBGA is 9.79, 4.69 and 2.22 respectively. The MCC is a balanced parameter that suggests a more compelling story for any model's classification. The high MCC value of 0.226 is exhibited by the CBGA model using the IMBD dataset, whereas the other two techniques get values of 0.477 and 0.1429, respectively. As a result, our suggested model successfully classifies sentiment. Furthermore, it is anticipated that the suggested model will categorise the other datasets with accuracy and dependability. The comparison of performance measures analysis for CBAO, CABO and CBGA models is shown in figure 8.

VI. CONCLUSION AND FUTURE WORK

We offer a sentiment analysis model that uses Bi- GRU, and an attention mechanism with a convolutional neural network to classify data using multiple labels in this study. This model employed many CNN channels rather than 1D CNN. Utilizing selected IMDB, Yelp 2015 and MR datasets, the model was tested. In terms of extracting features, the suggested hybrid model outperformed the single CNN and the bidirectional GRU. With comparison to existing baseline models, the suggested model produces the best classification model proposed, there are no syntactic structural features used in this investigation, traditional sentiment classification generally adds some syntactic structure features. Therefore, in incoming work, we will investigate how to combine traditional methods with methods of deep learning to increase classification accuracy even more.

REFERENCES

[1] B. Krawczyk, B. T. McInnes, and A. Cano, “Sentiment classification from multi-class imbalanced twitter data using binarization,” In Hybrid Artificial Intelligent Systems: 12th International Conference, pp. 26–37, 2017.

[2] Daniel Nu˜nez-Agurto, Walter Fuertes, Luis Marrone,and Mayra Macas, “Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions,” IAENG International Journal of Computer Science, vol. 49, no. 4, pp. 1002-1015, 2022.

[3] Cong-Cuong Le, P.W.C. Prasad, Abeer Alsadoon, L. Pham, and A. Elchouemi, “Text classification: Naïve bayes classifier with sentiment Lexicon,” IAENG International Journal of Computer Science, vol. 46, no. 2, pp. 141–148, 2019.

[4] E. Leopold and J. Kindermann, “Text categorization with support vector machines. How to represent texts in input space?,” Machine Learning, vol. 46, pp. 423–444, 2002.

[5] E. H. Han, G. Karypis, and V. Kumar, “Text categorization using weight adjusted k-nearest neighbor classification,” In Advances in Knowledge Discovery and Data Mining: 5th Pacific-Asia Conference, pp. 53–65, 2001.

[6] B. Yu, Z. B. Xu, and C. H. Li, “Latent semantic analysis for text categorization using neural network,” Knowledge-Based Systems, vol. 21, no. 8, pp. 900–904, 2008.

[7] Jie-Ming Yang, Zhi-Ying Liu, and Zhao-Yang Qu, “Clustering of words based on relative contribution for text categorization,” IAENG International Journal of Computer Science, vol. 40, no. 3, pp. 207–219, 2013.

[8] J. Hoffmann, O. Navarro, F. Kastner, B. Janßen, and M. Hubner, “A survey on CNN and RNN implementations,” In PESARO 2017: The Seventh International Conference on Performance, Safety and Robustness in Complex Systems and Applications, no. 3, pp. 1-7, 2017.

[9] Rosmina Bustami, Nabil Bessaih, Charles Bong, and Suhaila Suhaili, “Artificial Neural Network for Precipitation and Water Level Predictions of Bedup River,” IAENG International Journal of computer science, vol. 34, no. 2, pp. 228-233, 2007.

[10] W. Yin, K. Kann, M. Yu, and H. Schütze, “Comparative study of CNN and RNN for natural language processing,” In arXiv Preprint arXiv:1702.01923, 2017.

[11] J. P. Chiu and E. Nichols, “Named entity recognition with bidirectional LSTM-CNNs,” Transactions of the association for computational linguistics, vol. 4, pp. 357–370, 2016.

[12] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, “FastText.zip: Compressing text classification models,” In arXiv preprint arXiv:1612.03651, 2016.

[13] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, “Deep unordered composition rivals syntactic methods for text classification,” In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing, pp. 1681–1691, 2015.

[14] K. S. Tai, R. Socher, and C. D. Manning, “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks,” In arXiv preprint arXiv:1503.00075, 2015.

[15] X. Zhu, P. Sobihani, and H. Guo, “Long short-term memory over recursive structures,” In Proceedings of the32nd International Conference on Machine Learning, PMLR, vol. 37, pp. 1604–1612, 2015.

[16] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” In arXiv preprint arXiv:1408.5882, 2014.

[17] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” Advances in neural information processing systems, vol. 28, pp. 1-9, 2015.

[18] J. D. Prusa and T. M. Khoshgoftaar, “Designing a better data representation for deep neural networks and text classification,” In 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI), pp. 411–416, 2016.

[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 1480–1489, 2016.

[20] X. Zhou, X. Wan, and J. Xiao, “Attention-based LSTM network for cross-lingual sentiment classification,” In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 247–256, 2016.

[21] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “Disan: Directional Self-Attention Network for RNN/CNN-Free Language Understanding,” In Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1, pp. 5446-5455, 2018.

[22] Y. Liu, C. Sun, L. Lin, and X. Wang, “Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention,” In arXiv preprint arXiv:1605.09090, 2016.

[23] J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song and Q. Yang, “Large-scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN,” In Proceedings of the 2018 world wide web conference, pp. 1063–1072, 2018.

[24] L. Yao, C. Mao, and Y. Luo, “Graph Convolutional Networks for Text Classification,” In Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 1, pp. 7370–7377, 2019.

[25] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” In Proceedings of the 36th International conference on machine learning, vol. 97, pp. 6861–6871, 2019.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, , “Attention is all you need,” Advances in neural information processing systems, vol. 30, pp. 1-11, 2017.

[27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” In arXiv preprint arXiv:1412.3555, 2014.

[28] N. Wang, J. Wang, and X. Zhang, “YNU-HPCC at IJCNLP-2017 Task 4: Attention-based Bi-directional GRU Model for Customer Feedback Analysis Task of English,” In Proceedings of the 8th International Joint Conference on Natural Language Processing, pp. 174–179, 2017.

[29] L. Zhang, Y. Zhou, X. Duan, and R. Chen, “A Hierarchical multi-input and output Bi-GRU Model for Sentiment Analysis on Customer Reviews,” In IOP conference series: materials science and engineering, vol. 322, pp. 62007, 2018.

[30] J. Wu, K. Zheng, and J. Sun, “Text sentiment classification based on layered attention network,” In Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference, pp. 162–166, 2019.

[31] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, “Sentiment Analysis Using Gated Recurrent Neural Networks,” SN Computer Science, vol. 1, no. 74, pp. 1-13, 2020.

[32] S. K. Prabhakar, H. Rajaguru, and D.O. Won, “Performance Analysis of Hybrid Deep Learning Models with Attention Mechanism Positioning and Focal Loss for Text Classification,” Scientific Programming, vol. 2021,pp. 1-12, 2021.

[33] Y. Cheng, L. Yao, G. Xiang, G. Zhang, T. Tang, and L. Zhong, “Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism,” IEEE Access, vol. 8, pp. 134964–134975, 2020.

[34] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio,“Learning phrase representations using RNN encoder-decoder for statistical machine translation,” In arXiv Preprint arXiv:1406.1078, 2014.

[35] H. Yuan, X. U. Zhang, W. Niu, and K. Cui, “Sentiment analysis based on multi-channel convolution and bi-directional GRU with attention mechanism,” Journal of Chinese Information Processing, vol. 33, no. 10, pp. 109–118, 2019.

[36] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by

jointly learning to align and translate," In arXiv Preprint arXiv:1409.0473, 2014.

[37] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142–150, 2011.

[38] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," In arXiv Preprint cs/0409058, 2004.

[39] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A C-LSTM neural network for text classification," arXiv Preprint arXiv:1511.08630, 2015.

[40] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," In The Semantic Web: 15th International Conference, ESWC 2018, pp. 745–760, 2018.

[41] Y. Gao, W. Rong, Y. Shen, and Z. Xiong, "Convolutional neural network based sentiment analysis using Adaboost combination," In 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1333–1338, 2016.

[42] A. Hassan and A. Mahmood, "Deep learning approach for sentiment analysis of short texts," In 2017 3rd international conference on control, automation and robotics (ICCAR), pp. 705–710, 2017.

[43] S. Sohangir, D. Wang, A. Pomeranets, and T. M. Khoshgoftaar, "Big Data: Deep Learning for financial sentiment analysis," Journal of Big Data, vol. 5, no. 1, pp. 1–25, 2018.

[44] A. S. M. Alharbi and E. de Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," Cognitive Systems Research, vol. 54, pp. 50–61, 2019.

[45] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter," Future Generation Computer Systems, vol. 95, pp. 292–308, 2019.

[46] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," In 2020 39th Chinese control conference (CCC), pp. 7492–7497, 2020.

[47] M. Ghorbani, M. Bahaghighat, Q. Xin, and F. Özen, "ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing," Journal of Cloud Computing: Advances, Systems and Applications, vol. 9, no. 1, pp. 1–12, 2020.

[48] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," Concurrency and Computation: Practice and Experience, vol. 33, no. 23, pp. e5909, 2021.

[49] D. Nazarenko, I. Afanasieva, N. Golian, and V. Golian, "Investigation of the Deep Learning Approaches to Classify Emotions in Texts.," In COLINS: 5th International Conference on Computational Linguistics and Intelligent Systems, pp. 206–224, 2021.

[50] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, and A. Hussain, "Sentiment Analysis of Persian Movie Reviews Using Deep Learning," Entropy, vol. 23, no. 5, pp. 1-16, 2021.

[51] N. Gozuacik, C. O. Sakar, and S. Ozcan, "Social media-based opinion retrieval for product analysis using multi-task deep neural networks," Expert Systems with Applications, vol. 183, pp. 115388, 2021.

[52] V. Rupapara, F. Rustam, A. Amaar, P. B. Washington, E. Lee, and I. Ashraf, "Deepfake tweets classification using stacked Bi-LSTM and words embedding," PeerJ Computer Science, vol. 7, pp. e745, 2021.

[53] R. Kiran, P. Kumar, and B. Bhasker, "OSLCFit (organic simultaneous LSTM and CNN Fit): a novel deep learning based solution for sentiment polarity classification of reviews," Expert Systems with Applications, vol. 157, pp. 113488, 2020.

[54] L. Khan, A. Amjad, K. M. Afaq, and H. T. Chang, "Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media," Applied Sciences, vol. 12, no. 5, pp. 2694, 2022.

[55] Liu Ronghui, and Wei Xinhong, "Application of Improved Convolutional Neural Network in Text Classification," IAENG International Journal of Computer Science, vol. 49, no. 3, pp. 762-767, 2022.