IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 7, 2021, accepted February 14, 2021, date of publication March 9, 2021, date of current version March 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064830

A ConvBiLSTM Deep Learning Model-Based Approach for Twitter Sentiment Classification

SAKIRIN TAM[1,2], RACHID BEN SAID[2], AND Ö. ÖZGÜR TANRIÖVER[2]
[1]Information Technology Department, Fatoni University, 94160 Panttani, Thailand
[2]Computer Engineering Department, Ankara University, 06830 Ankara, Turkey

Corresponding author: Sakirin Tam (kirin.it@gmail.com)

ABSTRACT Being one of the most widely used social media tools, Twitter is seen as an important source of information for acquiring people's attitudes, emotions, views and feedbacks. Within this context, Twitter sentiment analysis techniques were developed to decide whether textual tweets express a positive or negative opinion. In contrast to lower classification performance of traditional algorithms, deep learning models, including Convolution Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM), have achieved a significant result in sentiment analysis. Although CNN can extract high-level local features efficiently by using convolutional layer and max-pooling layer, it cannot effectively learn sequence of correlations. On the other hand, Bi-LSTM uses two LSTM directions to improve the contexts available to deep learning algorithms, but Bi-LSTM cannot extract local features in a parallel way. Therefore, applying a single CNN or single Bi-LSTM for sentiment analysis cannot achieve the optimal classification result. An integrating structure of CNN and Bi-LSTM model is proposed in this study. ConvBiLSTM is implemented; a word embedding model which converts tweets into numerical values, CNN layer receives feature embedding as input and produces smaller dimension of features, and the Bi-LSTM model takes the input from the CNN layer and produces classification result. Word2Vec and GloVe were distinctly applied to observe the impact of the word embedding result on the proposed model. ConvBiLSTM was applied with retrieved Tweets and SST-2 datasets. ConvBiLSTM model with Word2Vec on retrieved Tweets dataset outperformed the other models with 91.13% accuracy.

INDEX TERMS Natural Language Processing, sentiment analysis, CNN, Bi-LSTM, Word2Vec, GloVe.

I. INTRODUCTION

Sentiment analysis (also known as opinion mining) refers to the use of text analysis and computational linguistic technique in NLP to identify, extract, and classify subjective information from unstructured text [1]. It aims to identify the polarity of sentences based on word clues extracted from the context of sentences [2]–[4]. Therefore, sentiment analysis is recognised as a significant technique to generate useful information from unstructured data sources such as tweets or reviews. In business, companies use sentiment analysis approach to understand their customer's feedback on their products or services. In politics, sentiment analysis is used as a decision-making tool to investigate the public reaction of political events. Social media platforms, including Twitter, Facebook, Instagram, blogs, reviews and news websites allow people to share widely their opinions and reviews. Twitter users have increased from 140 million in 2012 to 330 million active users in 2020 [5]. There are 145 million active users who publicly tweet on Twitter daily. These tweets contain hidden valued information that can be used to determine an author's attitude for a contextual polarity in the text [6], [7].

Even though statistical machine learning algorithms perform well for simpler sentiment analysis applications, these algorithms cannot be generalised to more complex text classification problems [8], [9]. On the other hand, deep learning models achieve significant results in sentiment analysis [10], speech recognition [11] and computer visions [12]. The two main deep learning algorithms that are widely used in sentiment analysis are Convolution Neural Network (CNN) and Recurrent Neural Network (RNN).

LeCun *et al.* [13] proposed to use CNN model to extract features from the text and learn local response from temporal or spatial data. He used the weight sharing approach in CNN model to reduce the computation complexity and training parameters. However, CNN cannot learn the sequence of correlation and effectiveness of CNN model is mainly based on the right selection of window size [15]. RNN is one of

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

deep learning models which are good for learning sequential model, but it cannot extract local features in a parallel way. RNN becomes a complementary approach to CNN because it can retain the sequence of information over time. Unfortunately, RNN greatly suffers from gradient exploding and vanishing problems [15]. These problems make RNN hard to train long-distance correlation in a sequence. Bidirectional-LSTM (Bi-LSTM) is one of RNN models which recently achieved remarkable results in text sentiment analysis. It contains two LSTM directions to improve the contexts available to the network. Bi-LSTM consists of both backward and forward hidden layers to allow the network to access both the preceding and succeeding context of sequence [16].

On the other hand, in text sentiment classification, the text is represented in the form of vectors, and generally in high dimensional space. When Bi-LSTM extracts contextual information from the features, it cannot put emphasis on the essential information [17]. In contrast to Bi-LSTM, CNN has a convolutional layer to extract feature of vectors and reduce its dimension. To overcome the limitation mentioned above, this study aims to propose a novel text classification deep learning model by integrating the structure of CNN and Bi-LSTM together. The new structure of ConvBiLSTM aims to solve the limitation of Bi-LSTM with the use of a convolutional layer in CNN model. The following present the proposed structure of ConvBiLSTM. The one-dimensional convolutional layer extracts n-gram features of input texts at different positions of sentences and reduces its dimensions. Then, these features are fed into Bi-LSTM to extract contextual information to classify sentiment results. The model is trained and evaluated on two datasets; one is tweets dataset that was crawled from Chicago city between 01 Sep 2019 and 31 Oct 2019 and the second one is SST-2 dataset. Word2Vec and GloVe embedding models were used for word embedding technique. The experimental results indicated that ConvBiLSTM model outperformed against other models and previous studies.

Our main contributions are summarised as follows:

- Word2Vec and Glove models were used as word embedding technique to present the tweets in the form of numeric values or vectors. These models are pre-train unsupervised word vectors that are trained with a large collection of words and can capture word semantics. The study applied these different word vector models to verify effectiveness of the model.
- ConvBiSLTM model based approach was proposed for text sentiment classification by integrating the structure of CNN and Bi-LSTM together. CNN model extracts local features from word embedding, Bi-LSTM captures long-distance dependencies, and finally these features are classified into the classification result.
- To confirm the effectiveness of ConvBiLSTM model the experimental results were benchmarked with other deep learning models, traditional machine learning models and experimental results of previous studies.

The structure of this paper is as follows: Section 2 provides theoretical background and relevant studies to text sentiment classification context, including the concept of CNN and Bi-LSTM. Details of ConvBiLSTM model construction are explained in Section 3. Complexity analysis of the model is explained in Section 4. Experiments of this study are presented in Section 5. Results of the experiment are discussed in Section 6. Section 7 discusses experimental benchmark with previous studies. Finally, the possible research improvement to the study is concluded and provided in Section 8.

II. BACKGROUND ON SENTIMENT CLASSIFICATION

This section introduces literature on existing approaches used in sentiment classification, including deep learning models for sentiment classification, CNN, Bi-LSTM and word embedding techniques.

A. DEEP LEARNING

Recently, deep learning algorithms have achieved remarkable results in natural language processing area. They represent data in multiple and successive layers. They have the ability to capture the syntactic features from sentences automatically without extra feature extracting techniques, which consume more resource and time. This is the reason why deep learning models have attracted attention from NLP researchers to explore sentiment classification.

By making use of a multi-layer perceptron structure in deep learning, CNN can learn high-dimensional, non-linear, and complex classification. As a result, CNN is used in many applications such as computer vision, image processing, and speech recognition [18], [19]. Kalchbrenner and Blunsom [20] designed Dynamic Convolution Neural Network (DCNN) model for text processing. Kim *et al.* [21] proposed English text classification by taking word vectors as input into CNN to get sentence-level classification. Even though CNN achieves good results in text classification, it mainly focuses on extracting local features and pays no attention to the context of words, which have much impact on the performance of text classification results [22], [23]. From this motivation of work, an integrated model of CNN with Bi-LSTM was proposed.

Automating the learning and expressing features in neural network enables RNN to integrate the adjacent location of information in NLP effectively. Long Short-Term Memory (LSTM) is one of RNN models [24] that can build a large-scale structure of neural network. LSTM makes good use of memory to avoid gradient problems in RNN [25]. In contrast to CNN and LSTM, RNN pays more attention to context of feature information and can fit into non-linear relations while retaining the sequential of text information [26], [27]. Also, Bidirectional RNN is another type of neural network models that is popular in text classification [28]. Bidirectional RNN works as the combination of two RNN models; backward and forward hidden layers,

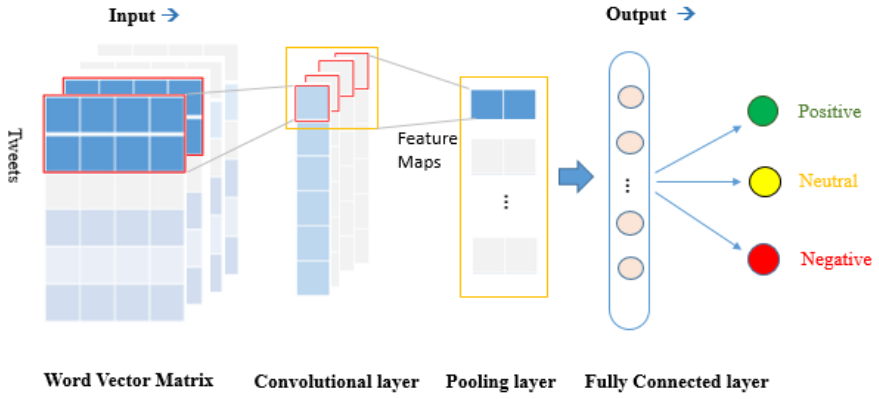

FIGURE 1. A generic architecture of CNN based for text classification [31].

to improve the performance of RNN neural network model. This approach can learn semantic information of words better because word semantics are correlated with preceding and succeeding information of the words.

B. CONVOLUTIONAL NEURAL NETWORK

CNN is a multi-layer feed-forward neural network which improves the error in backpropagation network (BP) and reduces computation time and complexity of BP [29], [30]. It is recently used for sentiment classification because it can recognise local features by using convolution kernel, and automatically learns these features for classification solution.

CNN model consists of three main layers; convolution layer, pooling layer, and fully connected layer [31]. Figure 1 shows the stages of CNN structure for text classification. Sentences are converted into a matrix of numbers and input to the convolutional layer. Each sentence consists of words or tokens, and each token is corresponded to a row or vector on the matrix table. These vectors are typically generated by embedding techniques such as the Word2Vec and GloVe model.

CNN model takes the input of vectors and extracts local feature using filters. The most computations of features are performed in convolutional layer which is the most important layer in CNN. Convolutional layer produces feature maps using a function called convolution kernel.

After the convolution operation, pooling layer extracts the most important features. The pooling layer calculates local sufficient statistics. This process allows the pooling layer to reduce feature dimensions, makes CNN achieve computational time and cost reduction, and prevents the model from overfitting problem. Finally, the fully connected layer produces a probability distribution to classify sentiment results.

C. LONG SHORT TERM MEMORY

RNN is one of deep learning algorithms which is mainly used in NLP to predict the next word base on previously given words in a sentence. RNN also uses back-propagation as other traditional neural networks. However, RNN suffers from gradient exploding and vanishing problems. These two problems make RNN hard to train and fine-tune parameters. These problems normally occur during back-propagation process. Long Short Term Memory (LSTM) is an RNN model to improve the problems mentioned above.

LSTM modifies the structure of RNN. It reconstructs RNN layer into a structure that contains a gate and a memory unit. The purpose of LSTM is to keep the information in the memory cell for further utilisation and update. With this new structure, LSTM solves the problems of gradient exploding and vanishing problem in RNN. Moreover, it is more promising to apply LSTM to solve sentiment analysis problems because its variants can capture long short-term dependencies.

D. BIDIRECTIONAL LSTM

Bi-LSTM is one of RNN algorithms to improve LSTM which has shortcomings of text sequence features. It solves the task of sequential modeling better than LSTM [32], [33]. In LSTM, information is flowed from backward to forward, whereas the information in Bi-LSTM flows in both directions; backward to forward and from forward to backward by using two hidden states. The structure of Bi-LSTM makes it a pioneer in sentiment classification because it can learn the context more effectively. Figure 2 shows the architecture of Bi-LSTM [34]. By utilising two ways of direction, input data of both preceding and succeeding sequence in Bi-LSTM are retained, unlike the standard RNN model that needs decay to include future data.

E. WORD EMBEDDING

Word embedding is an approach to represent words of text as a matrix of numeric values or vectors. It produces similar representation of vectors for words that have similar meanings. Word embedding approach is also called word vectorisation technique; each word is converted to one vector for the input of neural network. The mapping process is normally done in low-dimensional space but sometimes it depends on the size of vocabulary.

Word embedding is generally classified into probabilistic prediction and count-based approaches. The prediction

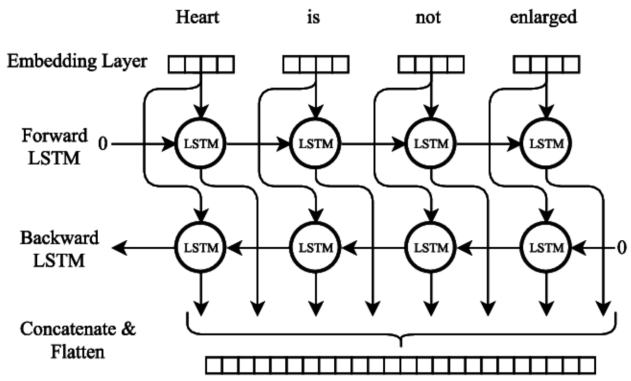

FIGURE 2. Architecture of a Bi-LSTM [34].

approach uses words composed from corpus to train the model. Word2Vec is one of the most outperformed probabilistic prediction approaches. It uses skip-gram and continuous bag of words (CBOW) methods to generate the word vector [35], [36]. With CBOW, a particular word is predicted based on its given neighbour, whereby the skip-gram predicts the neighbour word based on the given word.

Count-based approach uses frequency matrix of word co-occurrence to learn the vectors. GloVe model is mainly recognised in this approach [37]. GloVe model encodes the meaning of words based on the ratios of word-to-word co-occurrence probabilities. GloVe and Word2Vec models effectively generate word vectors for word similarity tasks [38].

III. PROPOSED APPROACH

This paper aims to develop a novel approach to improve sentiment classification on tweets by using the ConvBiLSTM model. In this section, structure of proposed model is discussed in detail. Figure 3 shows the overall structure of ConvBiLSTM model with five main phases as follows:

A. WORD VECTORISATION

In this phase, the network takes the input of raw text and segments into word or token one by one. Each token is converted into a vector of numeric values. Pre-trained word embedding models, including Word2Vec and GloVe, are used to generate word vector matrix. Word2Vec and GloVe models distinctly are used to observe the model performance. If each text of n words is represented as $T = \{w_1, w_2, \ldots, w_n\}$, then each word is converted into word vector of d dimension, the text of input is defined as:

$$T = \{w_1, w_2, \ldots, w_n\} \in R^{n*d} \tag{1}$$

Since the individual text of input have different lengths, its length needs to uniform (l). Its length was padded with zero-padding strategy. Text which has a length longer than the predefined length l will be truncated. But, if the text which has length shorter than l, zero padding will be added to the length. Therefore, all texts have the same dimension of matrix. Each text of l dimension is defined as follows:

$$T = \{w_1, w_2, \ldots, w_n\} \in R^{l*d} \tag{2}$$

B. CONVOLUTIONAL LAYER

CNN model is good at extracting the most important words from tweets or sentences [39] and the convolution layer is the main step in CNN model. The word vectors matrix $T \in R^{l*d}$ from word embedding layer are fed into one-dimensional convolution layer. In one-dimensional convolution layer, the convolution word vector matrix is calculated through N filters and width q of convolution kernel to construct the local feature of n-gram. Filter F_n, where $1 \leq n \leq N$ generates feature maps as follows:

$$c_i^n = f(w^n \otimes X_{i:i+w-1} + b^n) \tag{3}$$

Weight matrix of filter F_n is defined as $w \in R^{q*d}$, and b^n is the bias of filter F_n, d is word vector dimension, and $\otimes$ is convolution operation, $X_{i:i+q-1}$ indicates that filter F_n extracts feature $X_{i:i+q-1}$ from X_i, f is non-linear activation, and the output of feature map of F_n filter is c_i^n where i^{th} is element of c^n. In this study, RELU function was applied to non-linear activation f. For the sentence with length l, the following feature maps were obtained:

$$c = [c_1, c_2, \ldots, c_i, c_l] \tag{4}$$

C. MAX-POOLING LAYER

Once convolution operation produces feature maps, pooling layer then extracts the most important features $\widehat{c} = \max\{c\}$ to calculate the local sufficient statistics. One-dimensional max-pooling converts each kernel size of input into a single output of the maximum number to reduce or down-sample version of the input. This is the reason why CNN model effectively reduces the number of features to prevent overfitting, also reduces time and complexity of parameters.

D. BI-LSTM LAYER

In contrast to LSTM, Bi-LSTM allows the information to flow in both directions; backward to forward and from forward to backward by using two hidden states. This can help Bi-LSTM to learn the context better. By utilising these two-way directions, input data of both past and future information will be retained, whereby the standard RNN model needs decay to include future information. The principle implementation of Bi-LSTM is as; two opposite directions of LSTM network are connected to one output. The past information is obtained by forward LSTM state and the following information is obtained by backward LSTM state. This structure helps the network to retain preceding and succeeding information. The sequence output of the first layer in Bi-LSTM is the input of the second layer, and the sequence output of the second layer is the concatenation of the last unit output of forward and backward layers. After stacked Bi-LSTM layers, the final output is h:

$$h = [h_{forward}, h_{backward}] \tag{5}$$

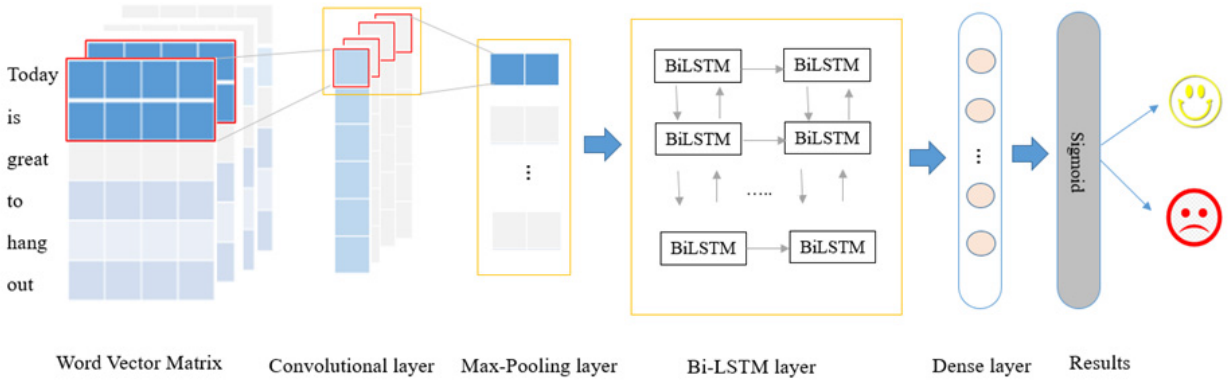

FIGURE 3. Architecture of proposed ConvBiLSTM model.

E. DENSE LAYER AND RESULT

Dense layer is used in the model to connect each input with every output by using weights. Sigmoid is a function used in the final layer to produce the output. It takes the average of the random results into 1 and 0 forms. The prediction result of sigmoid function is presented in Equation (6). the result of sentiment is classified into either 0 or 1 by using binary cross-entropy. In this study, 0 represents a negative sentiment and 1 represents a positive sentiment.

$$Y = sigmoid(wh + b) \tag{6}$$

F. REGULARISATION AND BATCH NORMALISATION

Overfitting is one of the most common problems in machine learning when the models train the data very well but fail to generalise on unseen data. Regularisation and batch normalisation are mainly used to avoid overfitting.

Regularisation is a technique which makes slight modification to the learning algorithm such that the model generalises better. It introduces additional information to lower the complexity of the model during training to prevent overfitting. The most popular regularisation is L2 and dropout. L2 regularisation is also known as weight decay or ridge regression, adds squared magnitude of coefficient as penalty to the loss function. Dropout is another technique to prevent overfitting and to generalise the network by randomly dropping a unit out (hidden and visible) during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on backward pass. In the study, 0.001 of L2 regularisation was set to the Bi-LSTM layer. Two dropout layers were added with dropout probability of 10% after max-pooling layer and before dense layer.

Batch normalisation is a technique used to normalise the inputs for each batch after dropout. This has the effect of stabilising the learning process and dramatically reducing the number of training epochs required to train deep networks. Batch normalisation layer was added after max-pooling layer to reduce internal covariate shift and to converge at a faster rate. Any parameter was not passed to batch normalisation layer.

IV. COMPLEXITY ANALYSIS

A. TIME COMPLEXITY ANALYSIS

Time complexity of an algorithm indicates how much time is required for the execution of a model. If the time complexity of a model is low, the execution requires less time to train and predict the model. Formula (7) denotes time complexity of CNN and Formula (8) denotes time complexity of Bi-LSTM.

$$Time \sim O(M^2 * Q^2 * Cin * Cout) \tag{7}$$
$$Time \sim O(M^2 * Q^2 * 2Cin * 2Cout) \tag{8}$$

where, M is output size of graph, Q is convolution kernel size, Cin is input channel numbers and $Cout$ is output channel numbers.

To compute time complexity of ConvBiLSTM, CNN was first used to compute the local features and then Bi-LSTM was used to compute the global features. The calculation of local feature is computed by using a filter, $n * q$,to convolute matrix of pair sentence with one step size. To get the largest feature, it requires scanning of n-row vectors to produce a set of local eigenvectors. Therefore, it needs n-1, and $O(n(n-1))$ of time complexity.

To compute global features of T tweets matrix, T tweets features are transmitted to Bi-LSTM via input gates, requiring $O(Cin)$ of time complexity. Therefore, by using novel structure of integrating CNN and Bi-LSTM greatly reduces time complexity against original Bi-LSTM.

B. SPATIAL COMPLEXITY ANALYSIS

Spatial complexity of an algorithm indicates the number of parameters in a model. By using formulas (3)-(5), CNN model can greatly extract and reduce the size of input features before feeding to Bi-LSTM for global features extraction. In addition, dropout layer was set before dense layer to update the weight independently on inherent characteristics. Therefore, the number of parameters on ConvBiLSTM is less

TABLE 1. Sample of tweets in datasets.

Tweets Dataset	
Positive Tweets	**Negative Tweets**
• **I'm pretty sure I was redesigning this**	• Can I borrow your dog sometime
• **Man I swear my life is perfect I could merch it**	• No one wants to see that little piece of shit I have to squint my eyes turn my head to the side
• **A person must really love you too teach you about the love of God at a young age**	• 20 seconds in, and Cabin Fever already has a dead dog.
• **Thoughts on kaepernick coming to the bears though Haha**	• Another day another meg
SST-2 Dataset	
Positive Tweets	**Negative Tweets**
• **That loves its characters and communicates something rather beautiful about human nature**	• Goes to absurd lengths
• **A smile on your face**	• Lend some dignity to a dumb story
• **The greatest musicians**	• For those moviegoers who complain that ` they don't make movies like they used to anymore

than traditional Bi-LSTM and spatial complexity is greatly reduced.

V. EXPERIMENT

The experiment was implemented to evaluate ConvBiLSTM model for text sentiment classification on tweets and SST-2 datasets. Tweets dataset is the dataset that was crawled from Twitter in Chicago. In this section, details of experimental setup, data pre-processing, hyper-parameters settings, and performance metrics were provided.

A. DATASET

To validate the proposed model, the proposed model was trained with two datasets. One is Tweets label sentiment analysis dataset that was collected from Twitter for two months of window time in Chicago, between 01September and 31October 2019. A total number of 797,324 tweets are classified into two classes. 1 is represented for positive tweets and 0 is represented for negative tweets. The total number of 475266 are negative class and 322058 are positive class.

The second dataset is the binary labeled version of Stanford Sentient Treebank (SST-2) dataset; there are 67349, 872, 1821 sentences of train, validation and test dataset, accordingly. SST-2 dataset was also used for model benchmarking with previous sentiment classification studies. The SST-2 dataset is available at ''https://www.kaggle.com/atulanandjha/stanford-sentiment-treebank-v2-sst2''.

Since the ratio of both classes in the dataset is not the same, SMOTE resampling method was applied to balance the datasets. The datasets were split into 60%, 20%, and 20% for training, testing, and validation dataset, accordingly, for the model execution, testing and verification. Table 1 shows the sample of tweets in the study datasets.

B. EXPERIMENTAL SETUP

Many tools and libraries are available to develop the deep learning models. The most preferable tool is Keras [40]. TensorFlow was used as backend of Keras because it supported GPU environment.

The following computer specification was experimented: NVIDIA GeForce RTX2080Ti 32GB of GPU, Intel Core i9-9900KF processor, 32 GB of RAM. In the study, the accuracy value was the main performance metric to compare the result with previous study easily. Other performance metrics were also used, such as Precision, Recall and F1 score, to evaluate the model.

C. DATA PREPROCESSING

Data cleaning is the most crucial step in NLP because the raw dataset always consists of words or symbols that computers do not understand. Therefore, data cleaning was performed to remove punctuation, special character, and stopping word from the datasets. Then, the tweets were lemmatise and tokenise to individual words. A total number of 243297 unique tokens were extracted from tweets dataset and 14255 unique tokens were extracted from SST-2 dataset. These unique tokens were the most common words that are used in the corpus for word vectorisation. To train text-based data, there is a need to represent the text into numerical value or vector. In this study, both Word2Vec and GloVe were applied to represent the text-based dataset separately and verify the effectiveness of our model.

D. HYPER-PARAMETERS SETTING

In many cases, the model may produce less accuracy or even produce overfitting or underfitting. To obtain high model performance, conducting hyper-parameters tuning is very critical. Therefore, the randomised search strategy was used to tune hyper-parameter and optimise the accuracy. Table 2 describes the hyper-parameters value in the proposed model.

E. METRICS

Standard performance evaluation is conducted to evaluate our proposed ConvBiLSTM model. The evaluation metrics below are the main metrics to evaluate our model.

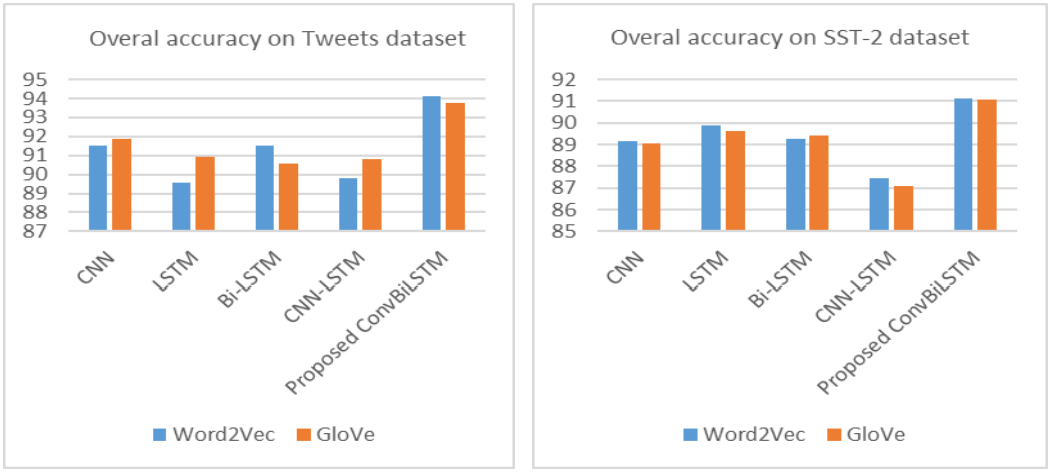

FIGURE 4. Overall accuracy of models based on word embedding techniques.

TABLE 2. Hyper-parameters setting.

Parameters	Values
Embedding dimension	Word2Vec =300 GloVe = 200
Kernel size	5
Filter	128
Pool size	2
Bi-LSTM output size	64
Kernel regularization	L2(0.001)
Weight constraints	Kernel constraint (max_norm=3)
Activation	Relu
(Recurrent) dropout	0.1
Batch size	128
Number of epoch	15
Batch normalization	yes
Loss function	Cross-entropy
Optimizer	Adam
Learning rate	0.001

TABLE 3. Classification matrix.

Categories	Model classification results	
	Positive (1)	**Negative (0)**
Positive (1)	True Positive (TP)	False Positive (FP)
Negative (0)	False Negative (FN)	True Negative (TN)

Evaluation metrics of the model can be performed as follows:

Accuracy:

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

Precision (p)

$$p = \frac{TP}{TP + FP} \tag{10}$$

Recall (r)

$$r = \frac{TP}{TP + FN} \tag{11}$$

F1-Score (f1):

$$f1 = \frac{2(TP + FP) + (TP + FN)}{TP} \tag{12}$$

VI. EXPERIMENTAL RESULTS

Based on the study with best hyper-parameter tuning, the proposed model was experimented with tweets dataset and SST-2 dataset against other deep learning models, including CNN, LSTM, Bi-LSTM, and CNN-LSTM. Figure 4 shows the overall accuracy of models based on Word2Vec and GloVe models. Word2Vec and GloVe are the two pre-train word vector methods that have received pioneer use in word vectorisation. The overall accuracy of experimental results showed that both Word2Vec and GloVe initialise word vectors for the datasets effectively.

The accuracy was slightly different on different word vector methods. For tweets dataset, GloVe method worked more effectively on CNN, LSTM, and CNN-LSTM, whereby Word2Vec produced outperformance on the proposed ConvBiLSTM model. Moreover, for SST-2, Word2Vec produced better accuracy than GloVe in almost all models, except for the Bi-LSTM model. Bi-LSTM with GloVe produced better result than Bi-LSTM with Word2Vec 0.17%. This proved that word vectorisation methods affected the accuracy of entire model.

Table 4 shows the overall result of different deep learning models on two datasets and two different word embedding techniques. The ConvBiLSTM model outperformed other deep learning models on both tweets and SST-2 datasets. The tweets dataset was observed to improve the model by 2.61% of overall accuracy, 3.16% of precision,0.29% of recall and 2.58 of F1 Score of all other models. In addition, the overall accuracy of ConvBiLSTM was improved by 2.62% better than CNN model, 4.56% better than LSTM

TABLE 4. Experimental results of deep learning models % is omitted and the best results are highlighted in bold.

Models	Dataset	Word2Vec				GloVe			
		p	r	F1	Accuracy	p	r	F1	Accuracy
CNN	Tweets dataset	90.03	88.88	88.6	91.51	89.19	90.92	89.26	91.89
LSTM		92.32	81.01	85.76	89.57	89.48	88.06	87.86	90.94
Bi-LSTM		86.21	**94.13**	89.24	91.52	84.42	93.99	88.1	90.57
CNN-LSTM		89.38	90.93	89.17	89.8	89.96	92.02	90.13	90.81
Proposed ConvBiLSTM		**95.49**	89.67	**91.82**	**94.13**	**90.5**	**94.42**	**91.81**	**93.76**
CNN	SST-2 dataset	90.34	90.46	90.14	89.18	91.69	88.5	89.77	89.05
LSTM		88.83	93.91	91.08	89.87	**93.27**	87.73	90.18	89.62
Bi-LSTM		92.39	88.08	89.94	89.25	88.05	93.96	90.68	89.42
CNN-LSTM		91.41	85.72	88.17	87.46	88.72	88.23	88.15	87.07
Proposed ConvBiLSTM		**94.6**	**94.33**	**92.08**	**91.13**	90.29	**94.2**	**92.01**	**91.06**

model, 2.61% better than Bi-LSTM, and 4.33% better than CNN-LSTM model. LSTM and CNN-LSTM achieved less accuracy. It was because LSTM lacks information on future context of large corpus of word in the network. For SST-2 dataset, the study model improved 1.26% of accuracy, 1% of precision, 0.37% of recall and 1.33% of F1 score of all other models. Also, the accuracy of proposed model was 1.95, 1.26, 1.72, 3.67 higher CNN, LSTM, Bi-LSTM and CNN-LSTM accordingly. In contrast to tweets dataset, LSTM model and Bi-LSTM models in SST-2 dataset achieved better result than CNN and CNN-LSTM model. It was because SST-2 has less dataset and made them easy to train.

It was observed that implementing sentiment analysis with CNN or Bi-LSTM alone could not reach effective results because the accuracy of CNN alone in the experiment was only 91.89% and the accuracy of experimental Bi-LSTM was 91.52% on Tweets dataset. Likewise, the accuracy of CNN and Bi-LSTM alone on SST-2 dataset was only 89.18% and 89.42%, respectively. This means that CNN and Bi-LSTM alone cannot achieve good result because CNN cannot learn sequence of correlation for long-term dependencies, and Bi-LSTM cannot capture local feature. When combining CNN with Bi-LSTM, the model can learn each word of tweets better because it has enough information of word context based on past and future context of word. Another observation was that Bi-LSTM achieved better result than LSTM because Bi-LSTM had the knowledge of preceding and succeeding information of text.

In addition, experiments were conducted with other traditional machine learning models to verify the model. The accuracy results in Figure 5 shows that the proposed model outperformed against others.

A running time record of ConvBiLSTM model and other deep learning models under the same circumstance are listed in Table 5. CNN model took 68 seconds for Word2Vec and 58 seconds for GloVe, in which the smallest running time in all models to complete a training epoch. Bi-LSTM took the longest running time, which were 187 seconds on Word2Vec and 154 seconds on GloVe. ConvBiLSTM took 159 seconds on Word2Vec and 140 seconds on GloVe, which was less than Bi-LSTM 28 and 14 seconds, respectively. The result proved that ConvBiLSTM model can improve the performance of sentiment without increasing training time.

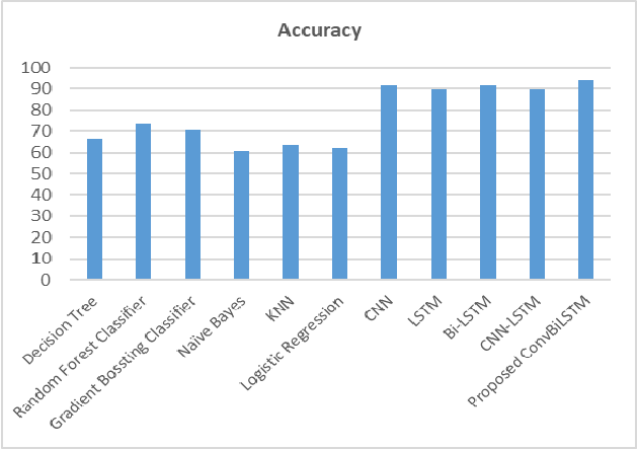

FIGURE 5. Accuracy of the proposed model against traditional models.

TABLE 5. Running time of each training epoch on tweets dataset.

Models	*Word2Vec*	*GloVe*
CNN	68	58
LSTM	146	120
Bi-LSTM	187	154
CNN+LSTM	135	135
ConvBiLSTM	159	140

VII. DISCUSSION

To verify the model with state-of-the-art, the experimental results were benchmarked with previous studies in text sentiment classification approaches on SST-2 dataset. Some of

TABLE 6. Accuracy of experimental results for sentiment classification on SST-2 dataset, % sign is omitted and the best results are highlighted in bold.

Models	Accuracy	Reported in
SVM	79.4	Lui et al. [41]
RAE	82.4	Socher et al.[42]
MV-RNN	82.9	Socher et al.[43]
RNTN	85.4	Socher et al.[44]
DCNN	86.8	Kalchbrenner et al.[45]
CNN-static	86.8	Kim [46]
CNN-non-static	87.2	Kim [46]
CNN-multichannel	88.1	Kim [46]
DRNN	86.6	Irsoy [47]
Multi-task-LSTM	87.9	Liu [48]
Tree-LSTM	86.9	Tai et al.[49]
C-LSTM	87.8	Zhou et al.[50]
BiLSTM-CRF	88.3	Tao et al.[51]
LSTM	86.4	Gang et al.[16]
BiLSTM	88.0	Gang et al.[16]
AC-BiLSTM	88.3	Gang et al.[16]
BiGru+CNN	85.40	Zang et al. [52]
CNN-BLSTM	89.7	Shen et al. [53]
ConvLSTMConv	89.02	Ghorbani et al. [54]
FEA-NN	73.31	Meng et al. [55]
ConvBiLSTM	**91.13**	**Our model**

these approaches were previously used in Kim's model [46]. From Table 6, it was observed that CNN-based models have better results than RAE, SVM and MR-RNN. These results proved the effectiveness of DNN approach and inspired many research studies to develop a sentiment classification by using deep learning algorithms.

Tao *et al.* [51] proposed BiLSTM-CRF and CNN model to improve on sentiment analysis. He applied sequential based model to extract and classify sentences into target expresses and used 1D-CNN for sentiment classification. His model achieved 88.3% of accuracy, which was better than the result in other DNN models. Gang and Jiaboa [16] proposed another deep learning model called AC-BiLSTM model with an accuracy of 88.3%. Gang used attention mechanism and convolutional layer to improve semantic understanding and accuracy of classification. Similar to Gang's model, Meng *et al.* [55] recommended feature enhanced attention CNN-BiLSTM (FEA-NN), who also used attention mechanism to improve feature extraction. However, BiLSTM-CRF, AC-BiLSTM and FEA-NN have more complicated structures which increased the complexity analysis. In contrast, the proposed model has less complicated structure which consists of just one convolutional layer and one BiLSTM layer but still achieved 2.83% higher than [16], [51] and 17.82 higher than [55].

Zang *et al.* [52] investigated a pipelining structure of BiGru and CNN for sentiment classification, which achieved an accuracy of 85.40%. However, BiGru-CNN ineffectively learned features when performing forward or backward propagation in the case of gradient explosion or vanishing. Opposite to BiGru-CNN, the proposed model, based on CNN and Bi-LSTM can not only effectively extract key features, but it can also improve accuracy; 5.73% higher than [52].

Shen *et al.* [53] suggested CNN+BLSTM model in six different experiments mainly 1, 2 or 3 CNN layers and a BLSTM layer respectively with pre-trained and without pre-trained word embedding. The result from Shen proved that 1CNN-BLSTM with pre-trained word embedding got the best accuracy of 89.7%. However, their experiment was conducted by using smaller size of word embedding of GloVe (50M dimensions) which led to less accuracy of 1.43% than the proposed model (200M dimensions). This result also verified the proposed observation upon applying different embedding model, which was discussed in Section 6, Figure 4.

Gorbani *et al.* [54] suggested another ConvLSTMConv model which extracts feature by CNN, learns contextual information by BiLSTM and its results are reused for CNN again to provide an abstract feature before applying to final dense layer. His model received remarkable result of 89.02% accuracy. However, the proposed model is simpler and has less complexity analysis but achieve higher accuracy 2.11% than in [54].

In summary, it is inferred that the proposed ConvBiLSTM model has simple structure but achieved outperform results on SST-2 dataset.

VIII. CONCLUSION

This paper aims to propose a novel sentiment classification model by integrating the structure of CNN with Bi-LSTM models. The proposed approach helps the model to classify text sentiment effectively by capturing both local and global dependencies in the contextual of sentences. The model is trained and evaluated on tweets dataset that were crawled from Chicago for two months of window time and SST-2 dataset. Both Word2Vecc and GloVe were utilised for word vectorisation. A total of 243297 and 14255 common tokens were represented by word vectors on both datasets. In the proposed model, CNN extracts text features and give its context information of text to Bi-LSTM. A crucial step was conducted to tune hyper-parameter to optimise the model. Finally, the model could classify text sentiment effectively on both datasets. The experiment result verified the feasibility and effectiveness of model.

For further research, the structure of ConvBiLSTM model could be modified to increase the performance of sentiment classification. Other NLP techniques such as POS tagging could improve the model accuracy. Another aspect that could be improved in the model is word embedding approaches. The experimental results proved that word representation could affect the accuracy of entire model. Therefore, an integrated word embedding approaches may produce a better feature extraction for the network.

REFERENCES

[1] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[2] B. Liu, *Sentiment Analysis and Opinion Mining* (Synthesis Lectures on Human Language Technologies), vol. 5, no. 1. San Rafael, CA, USA: Morgan & Claypool, 2012, pp. 1–167. Accessed: Nov. 1, 2020, doi: 10.2200/S00416ED1V01Y201204HLT016.

[3] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, pp. 271–278.

[4] P. D. Turney, "Thumbs up or thumbs down: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 417–424.

[5] Twitter. *Twitter Usage/Company Facts*. Accessed: Nov. 1, 2020. [Online]. Available: http://www.twitter.com

[6] S. Alami and O. Elbeqqali, "Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts," in *Proc. 10th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Oct. 2015, pp. 1–5.

[7] B. Liu, "Sentiment analysis: Mining opinions, sentiments, and emotions," *Comput. Linguistics*, vol. 42, no. 3, pp. 1–4, 2016.

[8] Q. Huang, R. Chen, X. Zheng, and Z. Dong, "Deep sentiment representation based on CNN and LSTM," in *Proc. Int. Conf. Green Informat.*, Aug. 2017, pp. 30–33.

[9] M. S. Neethu and R. Rajasree, "Sentiment analysis in Twitter using machine learning techniques," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[11] L. Brocki and K. Marasek, "Deep belief neural networks and bidirectional long-short term memory hybrid for speech recognition," *Arch. Acoust.*, vol. 40, no. 2, pp. 191–195, Jun. 2015.

[12] V. Campos, B. Jou, and X. Giró-i-Nieto, "From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction," *Image Vis. Comput.*, vol. 65, pp. 15–22, Sep. 2017.

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

[14] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI*, vol. 333, Nov. 2015, pp. 2267–2273.

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

[16] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019.

[17] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[18] F. Y. Zhou, L. P. Jin, and J. Dong, "Review of convolutional neural network," *Chin. J. Comput.*, vol. 1, pp. 35–38, Jan. 2017.

[19] Y. Li and H. B. Dong, "Text emotion analysis based on CNN and BiLSTM network feature fusion," *Comput. Appl*, vol. 38, no. 11, pp. 29–34, 2018.

[20] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," *Comput. Sci*, vol. 10, pp. 1–2, Jan. 2013.

[21] K. Kim, B.-S. Chung, Y. Choi, S. Lee, J.-Y. Jung, and J. Park, "Language independent semantic kernels for short-text classification," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 735–743, Feb. 2014.

[22] S. Liu, "Novel unequal clustering routing protocol considering based on network partition & distance for mobile education," *J. Netw. Comput. Appl.*, vol. 88, no. 15, pp. 1–9, 2017.

[23] D.-G. Zhang, S. Zhou, and Y.-M. Tang, "A low duty cycle efficient MAC protocol based on self-adaption and predictive strategy," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 828–839, Aug. 2018.

[24] C. Jin and W. Li, "Chinese word segmentation based on bidirectional LSTM neural network model," *Chin. J. Inf.*, vol. 32, no. 2, pp. 29–37, 2018.

[25] J. Chen, H. F. Li, and L. Ma, "Dimensional speech emotion recognition method based on multi-granularity feature fusion," *Signal Process*, vol. 33, no. 3, pp. 374–382, 2017.

[26] D.-G. Zhang, H.-L. Niu, and S. Liu, "Novel PEECR-based clustering routing approach," *Soft Comput.*, vol. 21, no. 24, pp. 7313–7323, Dec. 2017.

[27] D.-G. Zhang, Y.-M. Tang, Y.-Y. Cui, J.-X. Gao, X.-H. Liu, and T. Zhang, "Novel reliable routing method for engineering of Internet of vehicles based on graph theory," *Eng. Comput.*, vol. 36, no. 1, pp. 226–247, Dec. 2018.

[28] Y. X. Fan, J. F. Guo, and Y. Y. Lan, "Context-based deep semantic sentence retrieval model," *Chin. J. Inform. Sci*, vol. 31, no. 5, pp. 161–167, 2017.

[29] K. Simonyan and A. Zisserman, *Two-Stream Convolutional Networks for Action Recognition in Videos*. London, U.K.: Univ. of Oxford, 2014.

[30] F. Chollet, *Deep Learning With Python*. Shelter Island, NY, USA: Manning, 2017.

[31] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, Aug. 2020, doi: 10.1007/s10462-019-09794-5.

[32] W. Liu, P. Liu, Y. Yang, Y. Gao, and Y. Yi, "An attention-based syntax-tree and tree-LSTM model for sentence summarization," *Int. J. Performability Eng.*, vol. 13, no. 5, pp. 775–782, 2017.

[33] X. Niu, Y. Hou, and P. Wang, "Bi-directional LSTM with quantum attention mechanism for sentence modelling," in *Proc. 24th Int. Conf. Neural Inf. Process.* Guangzhou, China: Springer-Verlag, 2017, pp. 178–188.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[35] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013, *arXiv:1310.4546*. [Online]. Available: http://arxiv.org/abs/1310.4546

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781

[37] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[38] R. Lebret and R. Collobert, "Word emdeddings through Hellinger PCA," 2013, *arXiv:1312.5542*. [Online]. Available: http://arxiv.org/abs/1312.5542

[39] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 225–230.

[40] C. François. (2015). *Keras: The Python Deep Learning Library*. [Online]. Available: https://keras.io/

[41] Y. Liu, J. W. Bi, and Z. P. Fan, "A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm," *Inf. Sci*, vols. 394–395, pp. 38–52, Jul. 2017.

[42] R. J. Socher Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process., Assoc. Comput. Linguistics (ACL)*, Edinburgh, U.K., 2011, pp. 151–161.

[43] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn., Assoc. Comput. Linguistics (ACL)*, Jeju Island, South Korea, 2012, pp. 1201–1211.

[44] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment tree-bank," in *Proc. Conf. Empirical Methods Natural Lang. Process., Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, 2013, pp. 1631–1642.

[45] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Assoc. Comput. Linguistics (ACL)*, Baltimore, MD, USA, 2014, pp. 655–665.

[46] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process., Assoc. Comput. Linguistics (ACL)*, Doha, Qatar, 2014, pp. 1746–1751.

[47] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst., Neural Inf. Process. Syst. Found.*, Montreal, QC, Canada, 2014, pp. 2096–2104.

[48] P. Liu, X. Qiu, and H. Xuanjing, "Recurrent neural network for text classification with multi-task learning," in *Proc. 25th Int. Joint Conf. Artif. Intell., Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 2873–2879.

[49] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. Asian Fed. Natural Lang. Process. (ACL-IJCNLP)*, 2015, Beijing, China, pp. 1556–1566.

[50] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *Comput. Sci.*, vol. 1, no. 4, pp. 39–44, 2015.

[51] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.
[52] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018.
[53] Q. Shen, Z. Wang, and Y. Sun, "Sentiment analysis of movie reviews based on CNN-BLSTM," in *Proc. 2nd Int. Conf. Intell. Sci. (ICIS)*, Shanghai, China, 2017, pp. 164–171.
[54] M. Ghorbani, M. Bahaghighat, Q. Xin, and F. Özen, "ConvLSTM-Conv network: A deep learning approach for sentiment analysis in cloud computing," *J. Cloud Comput.*, vol. 9, no. 1, pp. 9–16, Dec. 2020, doi: 10.1186/s13677-020-00162-1.
[55] W. Meng, Y. Wei, P. Liu, Z. Zhu, and H. Yin, "Aspect based sentiment analysis with feature enhanced attention CNN-BiLSTM," *IEEE Access*, vol. 7, pp. 167240–167249, 2019.

SAKIRIN TAM was born in Kompong Cham, Cambodia, in 1985. He received the B.S. degree in information technology from Fatoni University, Pattani, Thailand, in 2010, and the M.S. degree in information technology management from University Technology Malaysia, Johor Bahru, Malaysia, in 2014. He is currently pursuing the Ph.D. degree in computer engineering with Ankara University, Ankara, Turkey. From 2010 to 2015, he was a Lecture with the Department of Information Technology, Fatoni University. His research interests include machine learning, deep leaning, natural language processing, sentiment analysis, data mining, green IT, and computer-based test.

RACHID BEN SAID received the B.S. degree in computer engineering from the Faculty of Science and Technology, Moulay Ismail University, Errachidia, Morocco, in 2012, and the M.S. degree in computer engineering from the National High School for Electricity and Mechanics, Hassan II University, Casablanca, Morocco, in 2014. He is currently pursuing the Ph.D. degree in computer engineering with Ankara University, Ankara, Turkey. His research interests include deep reinforcement learning, deep learning, machine learning, quantum machine learning, software defined networking, and the Internet of Things.

Ö. ÖZGÜR TANRIÖVER received the B.Sc. degree in computer engineering, the M.Sc. degree in science and technology policy, and the Ph.D. degree in information systems from Middle East Technical University, Ankara, Turkey. He was a Certified Information Systems Auditor (CISA) with the Department of Information Management, the Banking Regulation Agency. He is currently an Assistant Professor with the Department of Computer Engineering, Ankara University. His current research interests include software quality, information systems security, social informatics, and machine learning for prediction.