

A Tree-Structured Deep Learning Model for Improving Classification with Self-Adaption and Self-Learning



Nirmala Veluswamy^{*}, Jayanthi Boopathy^{*}

School of Computer Studies, RVS College of Arts and Science, Coimbatore 641402, India

Corresponding Author Email: nirmalavelusamy2018@gmail.com

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.111022>

ABSTRACT

Received: 5 January 2024

Revised: 5 May 2024

Accepted: 10 July 2024

Available online: 31 October 2024

Keywords:

dimensional sentiment analysis, valence-arousal, tree-based neural network, A-T-CNN-LSTM, attention strategy

Tree-structured deep learning classifier models are widely used in dimensional sentiment analysis for efficient feature representation and learning. From this perspective, an Adversarial Tree-structured Convolutional Neural Network with Long Short-Term Memory (A-T-CNN-LSTM) model was developed that adopts the Semantic-enabled Frequency-aware Generative Adversarial Network (SFGAN) to create more adversarial samples for predicting the Valence-Arousal (VA) of the texts or image classes. In contrast, an abrupt change in input data was not handled that impacts the model accuracy. Hence, this article proposes an Adversarial Attention T-CNN-LSTM (AA-T-CNN-LSTM) model to handle abrupt changes and uncertainties in the input data for dimensional sentiment analysis. This model aims to enhance self-adaptation and self-learning efficiency by integrating an attention strategy with the A-T-CNN-LSTM network. This model is constructed based on the SFGAN, CNN, LSTM and attention strategy layers. The CNN captures the spatial dependencies, whereas the LSTM captures the temporal dependencies of the given input data. The attention strategy layer is included after LSTM to adaptively control the proportion of spatial and temporal dependencies by emphasizing a few weights for final output vectors. Moreover, the prediction of VA ratings of the texts or image classes is achieved based on the final output vectors. Finally, the testing outcomes reveal that the AA-T-CNN-LSTM model on the Stanford Sentiment Treebank (SST) and CIFAR-10 datasets reaches an accuracy of 91.84% and 93.14%, respectively, contrasted with the state-of-the-art models.

1. INTRODUCTION

Deep neural models have gained widespread use in recent years, but tree-based approaches like Decision Trees (DTs) and Random Forests (RFs) are still commonly used for learning problems involving metadata [1]. These methods offer several benefits, such as handling a wide variety of attribute classes, insensitivity to data quantity and extracting features with minimal complexity. However, when input has spatial features, neural net designs like Deep Neural Network (DNN), CNN and Recurrent Neural Networks (RNNs) are recognized as alternatives. Such designs can create task-sensitive assumptions, obviating the requirement for domain-specific experts in specific situations like image classification [2]. However, building DNNs using tree-based methodologies in raw information is challenging due to traditional Fully Connected Networks (FCNs) lacking an inductive bias toward high-dimensional raw data [3]. Despite some efforts, there is no widely accepted neural design that can successfully adapt tree-based techniques and most systems rely on traditional DT learning in their loops. This challenges the application of neural designs in different settings, revealing a blind spot in our understanding of DNNs. As a result, CNNs have become the preferred architecture for large-scale image classification due to their ability to recognize objects based on their salient

features [4]. Supervised training is the core principle for training CNNs to identify images, as they are trained using a large collection of labeled images. The network recognizes individual features within previously labeled images, and every training sample is presented simultaneously during the learning process [5, 6]. Nevertheless, modern knowledge is gathered gradually over time, necessitating the creation of structures that can learn new data as it becomes available.

CNN unifies feature extraction and classification, updating the entire structure instantly when a part of the feature space is updated. However, iterative learning risks permanent knowledge loss. To achieve this, retrained CNNs must incorporate past data into the process of updating current information. Tree-CNN (T-CNN) [7] is an adaptive hierarchical network designed to address catastrophic forgetting and utilize previously-learned characteristics. The T-CNN network consists of hierarchically developing CNNs that take on new labels. Its hierarchical structure has been expanded to accommodate new labels as nodes, with branching based on feature sharing. Better categorization is achieved by focusing on leaves, which are allocated by early nodes into coarse super-classes. Pre-trained convolution layers can be easily included in the expanded network, but it struggles to learn task-related texts with the required features.

As a result, the T-CNN-LSTM model was developed, which

aims to increase granularity in dimensional sentiment analysis by utilizing the regional CNN and LSTM to predict VA sentence scores [8].

The CNN splits text into regions, assigning weights based on their importance in VA prediction, while the LSTM aggregates data to predict VA across all regions. The model considers local information and global relationships between texts, then locates task-related expressions and clauses using a region division mechanism. The model incorporates organized data into VA prediction but does not examine the effects of class variations, low prediction rates, or feature learning complexity. To tackle these problems, the A-T-CNN-LSTM model [9] was developed to improve VA prediction performance in sentiment analysis. It adopted the GAN to produce adversarial samples for limited dataset classes. Spectrum divergences between generated and real images were significant when using standard GANs. The SFGAN variant can prevent spectral loss during the discriminator training by integrating the Frequency-Aware Categorizer (FAC) for multi-domain confidence estimate and introducing semantic restricted sampling for image creation.

1.1 Problem statement

The A-T-CNN-LSTM model can solve optimization problems, obtaining the correct latent vector for GAN-generated image creation according to user specifications. But this model cannot handle abrupt changes in data features, or factors. Also, uncertainties were more complex, which causes poor granularity and prediction in dimensional sentiment analysis.

1.2 Major contributions of the manuscript

In this manuscript, the AA-T-CNN-LSTM model is proposed to handle abrupt changes and uncertainties in the input data for sentiment analysis. In this model, the attention strategy on the input dimensions forces the model to focus on decisive feature dimensions and ignore others. The main aim of integrating the attention strategy with the A-T-CNN-LSTM is to enhance self-adaptation and self-learning performance while abruptly changing the input data. In this AA-T-CNN-LSTM model, the A-T-CNN-LSTM is the main core network for classification. Both attention strategy and A-T-CNN-LSTM layers emphasize a few weights for feature vectors. First, the input vector is provided to the CNN layer followed by the LSTM to learn both spatial and temporal dependencies of the input vector, respectively. After the LSTM layer, the attention strategy layer is added to adaptively adjust the proportion of spatial and temporal dependencies for achieving effective prediction of VA ratings of the texts or image classes. Thus, it can handle uncertainty, or abrupt changes in the input data during the training phase for increasing the classification/prediction accuracy.

The following portions are prepared as follows: Section 2 presents the literature survey. Section 3 explains the AA-T-CNN-LSTM model. Section 4 exhibits the test outcomes. Section 5 summarizes the study and suggests possible improvements.

2. LITERATURE SURVEY

This section discusses a few previous research on tree-based

deep learning algorithms for different applications. Deep Fuzzy Tree (DFT) algorithm [10] was presented to handle the huge-scale hierarchical image classification with numerous classes by replacing the Softmax function in the deep learner. To set up the tree-learning and core classifiers, a new double fuzzy inter-class relationship measure was included. However, the improper number of nodes and tree depth reduced precision.

A neural machine translation was developed with a Gumbel Tree-LSTM-based encoder [11]. The authors focused on encoding an input phrase into a vector in an unsupervised-tree style and decoding it into a desired phrase. The learned tree representations were provided to the decoders as contextual data. Additionally, they developed a relation-gated LSTM model [12] to capture the correlation among different phrases. They also adopted a typed dependency Tree-LSTM, which utilizes the phrase dependency pattern and the dependency category to embed phrase significance into a dense vector.

A new Tree-RNN [13] was developed for topology-preserving deep graph embedding and learning. First, the trees were built from graphs and projected them into image space. Then, the TreeRNN was used to capture the patterns from the graph-tree pictures and categorize them into different graph categories. To better categorize network traffic, a Tree-RNN algorithm has been applied [14]. Each classifier in the tree's binary structure implements the small classification and separate rules for splitting traffic into distinct classes are provided. Time-related data characteristics were trained using the RNN and similarity between classes was estimated using cosine similarity. However, the databases were unbalanced, so the results were not particularly reliable.

An integrated ResNet and Tree-RNN model [15] was built for classifying neuron categories in rat brains. The Tree-RNN was developed to learn from the unorganized SWC-format document information and the ResNet was built to identify features from the organized 2D neuron pictures. These were integrated by fusing two feature vectors and classifying them by the novel 3-layer neural network. But it did not handle any changes in the neuron data.

The RF-CNN model [16] was created to use cardiac magnetic resonance in the diagnosis of coronary artery disease. Low-dimensional copies of the original high-dimensional images were sent to the CNNs so that they could automatically extract the relevant details. Then, the majority voting system was used to incorporate these features into the DTs for categorizing coronary artery disease. However, the number of features used in the DTs affected the RF efficiency.

An optimized hierarchical T-CNN algorithm [17] has been developed based on the sheep flock optimization to predict workload and increase power efficiency in cloud computing. But it was difficult to handle the sudden changes in the input data. A tree-structured model [18] was designed by removing the impact of variances among clusters. The tree-structured classifier was built by automatically creating a primary classification tree by a clustering scheme, which assembles identical subtypes and uses the pruning rule to remove unwanted groups from the tree order.

A Balanced binary Tree CNN (BT-CNN) [19] was designed that utilizes a binary tree-like architecture. It includes convolution and depthwise separable convolution group modules to optimize time and memory usage. The balanced approach of combining these modules aims to achieve optimal performance. However, it may struggle with sudden changes in features, resulting in reduced accuracy.

A Child-Sum Tree-LSTM model [20] was created to update nodes and edge vectors interactively, enhancing the learning of richer node representations. It attaches node embeddings to connected links and updates parent nodes with edge information. This process was repeated from bottom to top. It utilizes one constituent parser and one dependency parser to generate diverse formats. However, it was more effective for syntactic dependency tree structures than phrase trees. A tree-structured LSTM network [21] was used to analyze the glycan moiety, and a graph neural network was employed to integrate potential fragmentation pathways. However, its efficiency was hindered by insufficient training data.

2.1 Research gap

From this literature, it is addressed that the existing tree-structured deep learning models do not handle uncertainties or sudden changes in the input data or features during model training. This impacts the prediction/classification efficiency. Related to these models, the proposed AA-T-CNN-LSTM model can effectively handle uncertainties in the input data depending on the attention strategy for dimensional sentiment analysis, which predicts the VA scores of texts or classes of different images.

3. PROPOSED METHODOLOGY

This section explains the AA-T-CNN-LSTM model for dimensional sentiment analysis in detail. Figure 1 depicts a pipeline of this study, which involves two major processes: (i) SFGAN-based data augmentation and (ii) T-CNN-LSTM with an attention strategy for classification/prediction. First, the SFGAN [9] is applied to augment the training data by creating

more adversarial samples. After that, those data are used to train the Attention strategy integrated T-CNN-LSTM model. The trained model is later validated by the test data to predict VA ratings of texts or classify image classes.

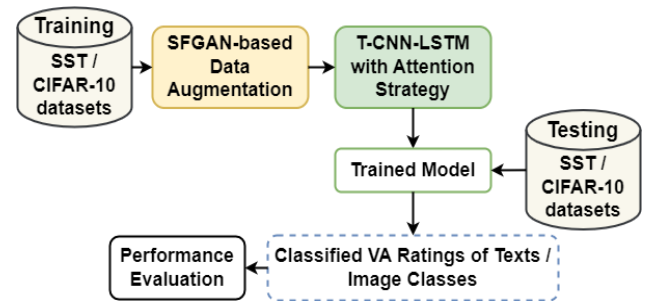


Figure 1. Pipeline of this study

3.1 Design of T-CNN-LSTM with an attention strategy

The design of the AA-T-CNN-LSTM model for classifying VA ratings of texts or image classes comprises two major components: (i) the SFGAN model [10] and (ii) the proposed T-CNN-LSTM with an attention strategy model. The T-CNN-LSTM splits every given vector into many regions rather than utilizing the entire input text or image to capture local n -gram traits in regions and long-range dependencies among regions. To split regions, a tree-structured region partition approach is adopted [9] such that the semantic features at various tree depths (regions) are used for classification tasks. A comprehensive description of the structure of the T-CNN-LSTM with an attention strategy is presented below. An overall structure of the T-CNN-LSTM with attention strategy is shown in Figure 2.

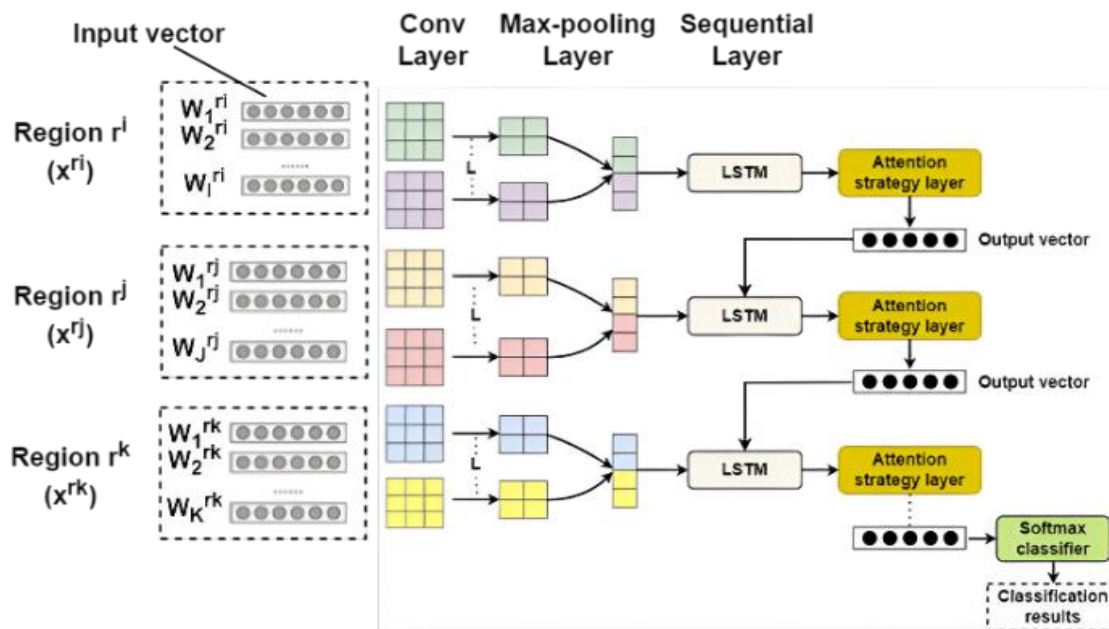


Figure 2. Structure of T-CNN-LSTM network with attention strategy

For each input vector, the T-CNN utilizes a portion of the input vector as a region to split the input into R regions, i.e., $r_1, \dots, r_i, r_j, r_k, \dots, r_R$. In every r , valuable sentimental traits are mined after the input vectors successively transfer via a convolution (conv) and max-pooling layers. Those local traits are successively united across R by the LSTM followed by the

attention strategy to create an output vector for VA classification.

Conv layer: In every region, the conv layer is utilized to mine local n -gram traits. Every input vector can be arranged in a region matrix $M \in \mathbb{R}^{d \times |V|}$, where $|V|$ denotes the data dimension of r and d denotes the size of the input vectors. For

instance, in Figure 2, the input vectors in the regions $r_i = \{v_1^{r_i}, v_2^{r_i}, \dots, v_l^{r_i}\}$, $r_j = \{v_1^{r_j}, v_2^{r_j}, \dots, v_j^{r_j}\}$ and $r_k = \{v_1^{r_k}, v_2^{r_k}, \dots, v_k^{r_k}\}$ are united to create the region matrices m^{r_i} , m^{r_j} and m^{r_k} . In all regions, L convolutional kernels are used to capture local n -gram traits. In a window of ω input vectors $m_{n:n+\omega-1}$, a kernel F_l ($1 \leq l \leq L$) produces the feature map y_n^l in this manner:

$$y_n^l = f(W^l \circ m_{n:n+\omega-1} + b^l) \quad (1)$$

In Eq. (1), \circ denotes the convolutional operator, b^l is the weight matrix and bias related to F_l , ω denotes the kernel length, d denotes the input vector dimension and f refers to the Rectified Linear Unit (ReLU) activation function. If a kernel slowly traverses from $m_{1:\omega-1}$ to $m_{N:\omega-1:N}$, the output feature maps $y^l = \{y_1^l, \dots, y_{N-\omega+1}^l\}$ of F_l . For varying input vector lengths in R , y^l can contain various sizes for various input vectors. So, the highest length of the CNN input in R is defined as the size N . When the input length is lower than N , no vectors can be added. As illustrated in Figure 2, all conv layers consider their input as a region vector to L distinct kernels and creates feature maps $Y = \{y^1, \dots, y^L\} \in \mathbb{R}^{(N-\omega+1) \times L}$.

Max-pooling layer: Max-pooling reduces the result of the conv layer by applying a max operation with a pooling dimension ρ to the outcome of all kernels. This helps preserve important details by mining local dependencies within multiple regions. The resulting M is then flattened to a vector and passed to the LSTM network.

LSTM layer: To extract long-range dependencies across R , the LSTM network successively combines all region vectors with the input vectors. It consists of an input gate, forget gate and output gate. The input gate regulates the present input, the output gate regulates the present result, and the forget gate regulates the past state. Eq. (2) calculates the forget gate, Eqs. (3)-(4) calculate the input gate, Eq. (5) updates the memory unit, and Eqs. (6)-(7) calculate the output gate.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \text{Sigmoid}(W_i[h_{t-1}, x_t] + b_i) \quad (3)$$

$$c'_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (4)$$

$$c_t = c_{t-1} \cdot f_t + i_t \cdot c'_t \quad (5)$$

$$o_t = \text{Sigmoid}(W_o[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = \tanh(c_t) \cdot o_t \quad (7)$$

In Eqs. (2)-(7), h_{t-1} and h_t denote the result at time $t-1$ and t , correspondingly, x_t represents the input at t , W and b are the weight matrix and bias, respectively, f_t indicates the result of the forget gate, i_t denotes the result of input gate, o_t , h_t are the input and result of output gate, correspondingly, c'_t , c_t represent the present state of input and output memory units, respectively. After the LSTM cell sequentially traverses through each region, the final hidden state h_t is defined as the output representation, which is given to the attention strategy layer.

Attention strategy layer: The attention strategy for input lengths intends to direct the model's focus toward the most important feature dimensions while disregarding others. In the attention layer, two processes are performed: (i) the output of

the upstream layer is merged and (ii) the decisive feature dimensions are chosen for classification. An attention strategy is utilized to reallocate the weights of feature representations. The attention strategy initially determines the final hidden state and attention score vectors depending on the input data from multiple channels of LSTM layers. The scores of representations are obtained by the dot-product function. The attention weight coefficient is calculated by:

$$e_t = u \tanh(wh_t + b) \quad (8)$$

$$a_t = \frac{e^{e_t}}{\sum_{i=1}^n e^{e_i}} \quad (9)$$

$$s_t = \sum_{i=1}^n e_i a_t \quad (10)$$

In Eqs. (8)-(10), e_t is the decision features of the LSTM's result h_t at t , u and w denote the weight coefficients, b represents the bias, a_t indicates the normalized weight coefficient and s_t represents the result of attention at t .

Softmax layer: In the last layer of the T-CNN-LSTM, the Softmax classifier is applied to classify the output vector from the LSTM into valence and arousal of the texts, or different classes of images.

The T-CNN-LSTM with the attention strategy is trained by reducing the mean squared error between the classified y and observed y . For a training set of the input matrix $X = \{x^1, \dots, x^m\}$ and their VA ratings set, or image classes $y = \{y^1, \dots, y^m\}$, the loss function is described by:

$$L(X, y) = \frac{1}{2m} \sum_{i=1}^m \|h(x^i) - y^i\|^2 \quad (11)$$

In the training stage, a backpropagation procedure with an Adam optimizer is utilized to fine-tune the model parameters. Thus, the AA-T-CNN-LSTM is trained and tested for dimensional sentiment analysis.

4. EXPERIMENTAL RESULTS

The efficiency of the AA-T-CNN-LSTM is compared with the existing models in this section. The experiment is carried out on a computer armed with Intel® Core™ i5-4210 CPU@2.80GHz, 8GB RAM, 1TB HDD under Windows 10 64-bit OS. This experiment uses two distinct datasets: Stanford Sentiment Treebank (SST) [22] and CIFAR-10 [23].

The SST dataset consists of 11,855 single sentences from movie reviews, parsed into 215,154 unique phrases. Each phrase is labeled as very negative, negative, neutral, positive, or very positive. For this study, 8,555 sentences (1711 for each class) are used for training and 3,300 (660 for each label) for testing.

The CIFAR-10 dataset contains 60000 color images of dimension 32×32 in 10 different categories, with 6000 images per category. Of these, 50000 images (5000 for each class) are used for learning and 10000 (1000 for each class) are used for testing.

To conduct a fair comparative study, all the existing models (e.g., T-CNN [7], T-CNN-LSTM [8], A-T-CNN-LSTM [9], Tree-RNN [14] and RF-CNN [16]) and proposed AA-T-CNN-LSTM are implemented in Python 3.7.8 software. The following performance evaluation metrics are determined:

Accuracy: It is a proper classification of VA of texts or image class among the overall examples tested.

$$Acc = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \quad (12)$$

In Eq. (12), TP is the total +ve texts exactly categorized as +ve, TN is the total -ve texts exactly categorized as -ve, FP is the total -ve texts incorrectly classified as +ve and FN is the total +ve texts incorrectly classified as -ve.

Precision: It is calculated by:

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

Recall: It is calculated by:

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

F-measure: It is determined by:

$$F - measure = 2 \times \frac{Precision \cdot Recall}{Precision+Recall} \quad (15)$$

Mean Absolute Error (MAE): It measures the average magnitude of the errors in a set of predictions. It is calculated as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (16)$$

In Eq. (16), n represents the number of images/sentences, y_i is the true label and \hat{y}_i is the classified label.

Receiver Operating Characteristic (ROC) Curve: It is the relationship between the FP rate and TP rate.

4.1 Performance analysis for SST dataset

Table 1 presents the confusion matrix for the AA-T-CNN-LSTM model using the SST dataset. Additionally, Figure 3 shows the performance of AA-T-CNN-LSTM during training and testing phases using the SST dataset.

Table 1. Confusion matrix for AA-T-CNN-LSTM model using the SST dataset

		True Labels				
		0	1	2	3	4
Classified labels	0	580	8	15	9	15
	1	20	629	17	10	14
	2	18	9	600	13	12
	3	19	8	13	617	13
	4	23	6	15	11	606

*Note: 0 – Very negative; 1 – Negative; 2 – Neutral; 3 – Positive; 4 – Very positive

In Figure 4, a comparison of the AA-T-CNN-LSTM model against existing models tested using the SST dataset is shown. It is observed that the precision of the AA-T-CNN-LSTM is improved by 15.58%, 14.06%, 11.81%, 7.45% and 1.05%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM, respectively. The recall of the AA-T-CNN-LSTM is better by 15.49%, 14.16%, 11.53%, 7.26% and 1.75%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively. The f-measure of the AA-T-CNN-LSTM is enhanced by 15.52%, 14.11%, 11.67%, 7.35% and 1.18%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively. Also, the

accuracy of the AA-T-CNN-LSTM is increased up to 15.71%, 14.53%, 11.85%, 7.65% and 1.91%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively.

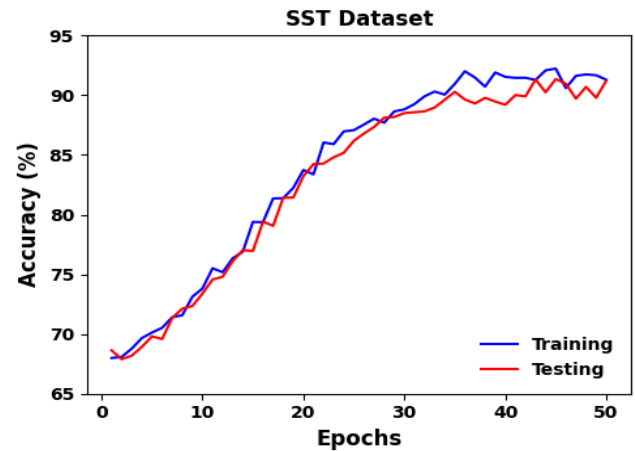


Figure 3. Performance of AA-T-CNN-LSTM during training and testing on SST dataset

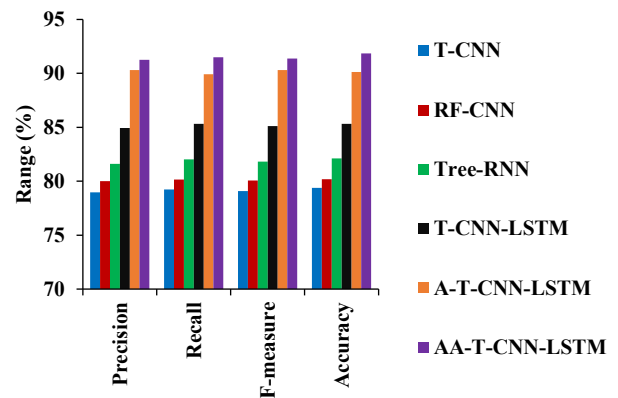


Figure 4. Comparison of AA-T-CNN-LSTM against existing models on SST dataset

Figure 5 illustrates the MAE for various tree-structure deep learning models on the SST dataset. The MAE of AA-T-CNN-LSTM is 60.45%, 58.81%, 54.39%, 44.45% and 17.41% less than the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively.

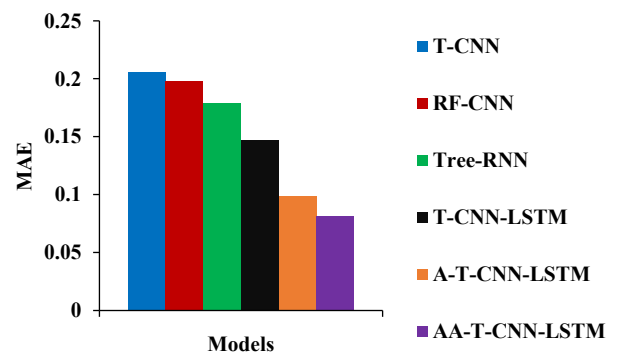


Figure 5. MAE of AA-T-CNN-LSTM model against existing tree-structured deep learning models on SST dataset

Figures 6 and 7 display the ROC curve and precision vs. recall curve, respectively, for different models on the SST dataset. The AA-T-CNN-LSTM network outperformed other models by enhancing self-learning and self-adaptation through the attention strategy in A-T-CNN-LSTM.

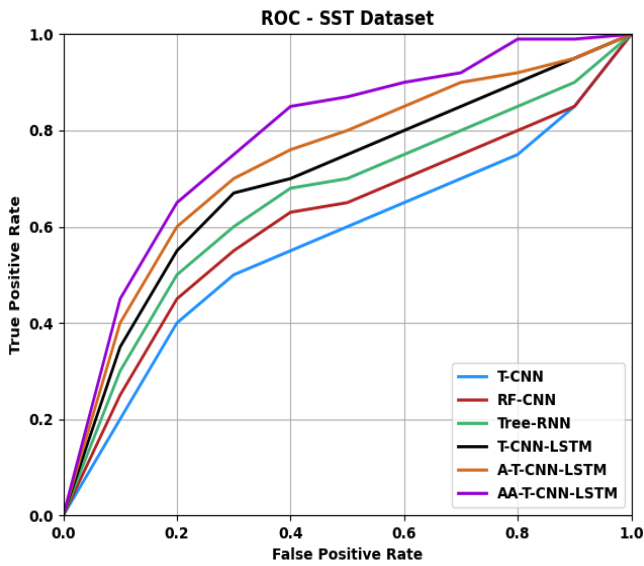


Figure 6. ROC curve of AA-T-CNN-LSTM against existing models on SST dataset

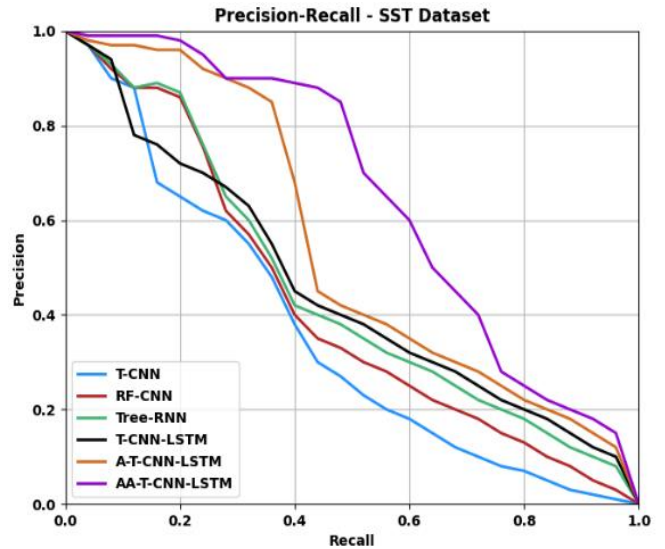


Figure 7. Precision vs. recall curve of AA-T-CNN-LSTM against existing models on SST dataset

4.2 Performance analysis using CIFAR-10 dataset

Table 2 presents the confusion matrix for the AA-T-CNN-LSTM model using the CIFAR-10 dataset. Additionally, Figure 8 shows the performance of AA-T-CNN-LSTM during training and testing phases using the CIFAR-10 dataset.

Table 2. Confusion matrix for AA-T-CNN-LSTM model using the CIFAR-10 dataset

Classified labels	True Labels									
	0	1	2	3	4	5	6	7	8	9
0	935	17	10	7	3	11	4	8	2	0
1	10	893	8	5	3	13	0	5	3	10
2	8	6	900	15	4	9	2	10	5	6
3	5	8	15	931	6	12	0	8	0	9
4	6	18	10	4	965	21	6	4	6	8
5	15	5	6	13	2	890	3	13	4	5
6	5	7	9	4	7	8	980	9	0	7
7	7	28	11	7	2	15	0	905	5	6
8	0	10	20	8	0	11	2	20	970	4
9	9	8	11	6	8	10	3	18	5	945

*Note: 0 – Airplane; 1 – Automobile; 2 – Bird; 3 – Cat; 4 – Deer; 5 – Dog; 6 – Frog; 7 – Horse; 8 – Ship; 9 – Truck

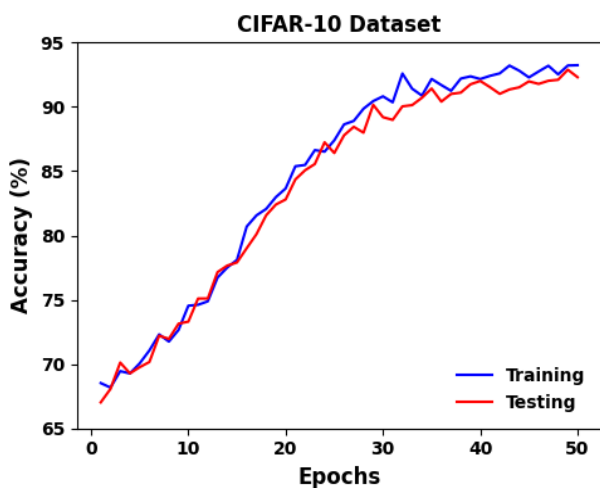


Figure 8. Performance of AA-T-CNN-LSTM during training and testing on CIFAR-10 dataset

Figure 9 shows the comparison of the AA-T-CNN-LSTM against existing models tested using the CIFAR-10 dataset. It is noted that the precision of the AA-T-CNN-LSTM is improved by 14.03%, 12.81%, 10.85%, 7.64% and 1.73%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM, respectively. The recall of the AA-T-CNN-LSTM is better by 14.45%, 12.97%, 10.91%, 8.15% and 2.21%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively.

The f-measure of the AA-T-CNN-LSTM is enhanced by 14.23%, 12.88%, 10.87%, 7.9% and 1.97%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM, respectively. Also, the accuracy of the AA-T-CNN-LSTM is increased up to 14.51%, 12.88%, 11.01%, 8.3% and 2.35%, compared to the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively.

Figure 10 illustrates the MAE for various tree-structure deep learning models on the CIFAR-10 dataset. The MAE of AA-T-CNN-LSTM is 63.24%, 60.78%, 57.39%, 51% and

23.78% less than the T-CNN, RF-CNN, Tree-RNN, T-CNN-LSTM and A-T-CNN-LSTM models, respectively.

Figure 11 and Figure 12 display the ROC curve and precision vs. recall curve, respectively, for different models on the CIFAR-10 dataset. Thus, it is realized that the AA-T-CNN-LSTM model on the CIFAR-10 dataset achieves superior efficiency in classifying VA ratings of texts or image classes compared to the other models by improving the self-learning and self-adaptation abilities based on the attention strategy in A-T-CNN-LSTM.

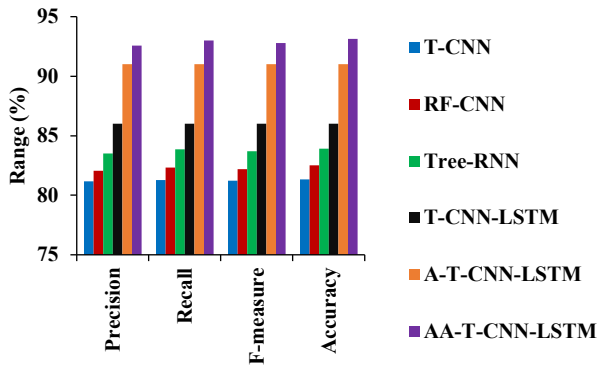


Figure 9. Comparison of AA-T-CNN-LSTM against existing models on CIFAR-10 dataset

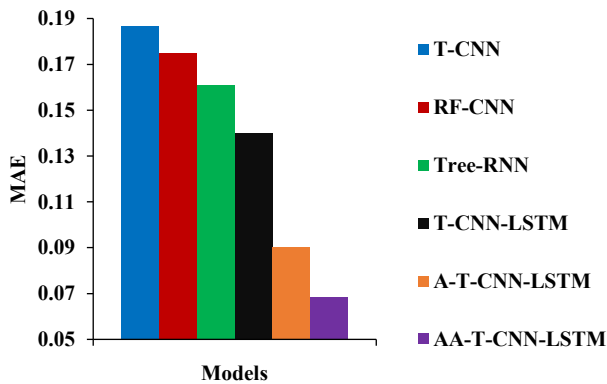


Figure 10. MAE of AA-T-CNN-LSTM against existing models on CIFAR-10 dataset

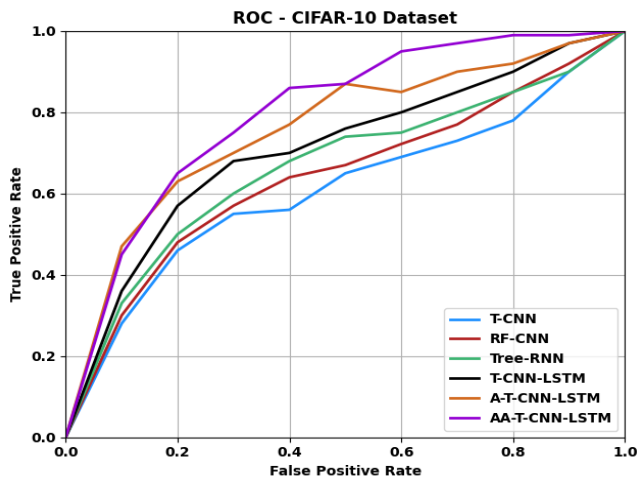


Figure 11. ROC curve of AA-T-CNN-LSTM against existing models on CIFAR-10 dataset

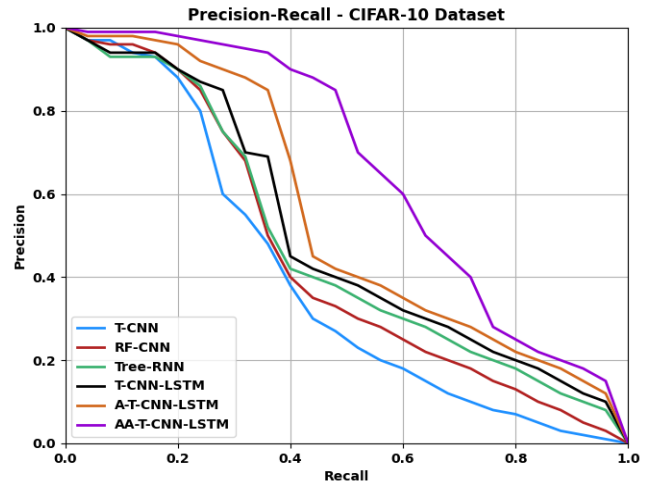


Figure 12. Precision vs. recall curve of AA-T-CNN-LSTM against existing models on CIFAR-10 dataset

5. CONCLUSION

In this study, the AA-T-CNN-LSTM model was developed to achieve dimensional sentiment analysis, which classifies the VA ratings of texts, or image classes while handling uncertainties in the given data. First, the training data was augmented by creating the adversarial samples using the SFGAN model. Then, the T-CNN-LSTM network with the attention strategy was trained using those data for classifying the VA ratings of texts, or image classes. Using the attention strategy, the fraction of spatial and temporal dependencies captured by the CNN and LSTM layers, respectively was adjusted by emphasizing a few weights for the final output vector. Extensive experiments were carried out using the SST and CIFAR-10 databases to assess the efficiency of the AA-T-CNN-LSTM against existing models. The test results proved that the AA-T-CNN-LSTM on the SST and CIFAR-10 datasets achieved 91.84% and 93.14% accuracy, respectively compared to the conventional tree-structured deep learning models.

REFERENCES

- [1] Ellen, J.S., Graff, C.A., Ohman, M.D. (2019). Improving plankton image classification using context metadata. *Limnology and Oceanography: Methods*, 17(8): 439-461. <https://doi.org/10.1002/lom3.10324>
- [2] Teng, C.F., Wu, A.Y.A. (2020). Convolutional neural network-aided tree-based bit-flipping framework for polar decoder using imitation learning. *IEEE Transactions on Signal Processing*, 69: 300-313. <https://doi.org/10.1109/TSP.2020.3040897>
- [3] Abpeikar, S., Ghatee, M., Foresti, G.L., Micheloni, C. (2020). Adaptive neural tree exploiting expert nodes to classify high-dimensional data. *Neural Networks*, 124: 20-38. <https://doi.org/10.1016/j.neunet.2019.12.029>
- [4] Rawat, W., Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9): 2352-2449. https://doi.org/10.1162/neco_a_00990
- [5] Khan, A., Sohail, A., Zahoor, U., Qureshi, A.S. (2020). A survey of the recent architectures of deep

- convolutional neural networks. *Artificial Intelligence Review*, 53(8): 5455-5516. <https://doi.org/10.1007/s10462-020-09825-6>
- [6] Qi, G.J., Luo, J. (2020). Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 2168-2187. <https://doi.org/10.1109/TPAMI.2020.3031898>
- [7] Roy, D., Panda, P., Roy, K. (2020). Tree-CNN: A hierarchical deep convolutional neural network for incremental learning. *Neural Networks*, 121: 148-160. <https://doi.org/10.1016/j.neunet.2019.09.010>
- [8] Wang, J., Yu, L.C., Lai, K.R., Zhang, X. (2019). Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 581-591. <https://doi.org/10.1109/TASLP.2019.2959251>
- [9] Velusamy, N., Boopathy, J. (2023). An enhancement of tree-structured deep learning classification through semantic enabled frequency aware data augmentation. *International Journal of Intelligent Engineering & Systems*, 16(6): 650-658. <https://doi.org/10.22266/ijies2023.1231.54>
- [10] Wang, Y., Hu, Q., Zhu, P., Li, L., Lu, B., Garibaldi, J.M., Li, X. (2020). Deep fuzzy tree for large-scale hierarchical visual classification. *IEEE Transactions on Fuzzy Systems*, 28(7): 1395-1406. <https://doi.org/10.1109/TFUZZ.2019.2936801>
- [11] Su, C., Huang, H., Shi, S., Jian, P., Shi, X. (2020). Neural machine translation with Gumbel Tree-LSTM based encoder. *Journal of Visual Communication and Image Representation*, 71: 102811. <https://doi.org/10.1016/j.jvcir.2020.102811>
- [12] Kleenankandy, J., Nazeer, K.A.A. (2020). An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies. *Information Processing & Management*, 57(6): 102362. <https://doi.org/10.1016/j.ipm.2020.102362>
- [13] Lyu, Y., Li, M., Huang, X., Guler, U., Schaumont, P., Zhang, Z. (2021). TreeRNN: Topology-preserving deep graph embedding and learning. In *IEEE 25th International Conference on Pattern Recognition, Milan, Italy*, pp. 7493-7499. <https://doi.org/10.1109/ICPR48806.2021.9412808>
- [14] Ren, X., Gu, H., Wei, W. (2021). Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Systems with Applications*, 167: 1-9. <https://doi.org/10.1016/j.eswa.2020.114363>
- [15] Zhang, T., Zeng, Y., Zhang, Y., Zhang, X., Shi, M., Tang, L., Zhang, D., Xu, B. (2021). Neuron type classification in rat brain based on integrative convolutional and tree-based recurrent neural networks. *Scientific Reports*, 11(1): 7291. <https://doi.org/10.1038/s41598-021-86780-4>
- [16] Khozeimeh, F., Sharifrazi, D., Izadi, N.H., Joloudari, J.H., Shoeibi, A., Alizadehsani, R., Tartibi, M., Hussain, S., Sani, Z.A., Khodatars, M., Sadeghi, D., Khosravi, A., Nahavandi, S., Tan, R.S., Acharya, U.R., Islam, S.M.S. (2022). RF-CNN-F: Random forest with convolutional neural network features for coronary artery disease diagnosis based on cardiac magnetic resonance. *Scientific Reports*, 12(1): 11178. <https://doi.org/10.1038/s41598-022-15374-5>
- [17] Selvan Chenni Chetty, T., Bolshev, V., Shankar Subramanian, S., Chakrabarti, T., Chakrabarti, P., Panchenko, V., Yudaev, I., Daus, Y. (2023). Optimized hierarchical tree deep convolutional neural network of a tree-based workload prediction scheme for enhancing power efficiency in cloud computing. *Energies*, 16(6): 2900. <https://doi.org/10.3390/en16062900>
- [18] Cai, Q., Niu, L., Shang, X., Ding, H. (2023). A self-supervised tree-structured framework for fine-grained classification. *Applied Sciences*, 13(7): 4453. <https://doi.org/10.3390/app13074453>
- [19] Chauhan, S., Cheruku, R., Reddy Edla, D., Kampa, L., Nayak, S.R., Giri, J., Mallik, S., Aluvala, S., Boddu, V., Qin, H. (2024). BT-CNN: A balanced binary tree architecture for classification of brain tumour using MRI imaging. *Frontiers in Physiology*, 15: 1349111. <https://doi.org/10.3389/fphys.2024.1349111>
- [20] Wang, L., Cao, H., Yuan, L. (2024). Child-Sum (N2E2N) Tree-LSTMs: An interactive child-sum tree-LSTMs to extract biomedical event. *Systems and Soft Computing*, 6: 200075. <https://doi.org/10.1016/j.sasc.2024.200075>
- [21] Yang, Y., Fang, Q. (2024). Prediction of glycopeptide fragment mass spectra by deep learning. *Nature Communications*, 15(1): 2448. <https://doi.org/10.1038/s41467-024-46771-1>
- [22] Sentiment Analysis. <https://nlp.stanford.edu/sentiment/treebank.html>.
- [23] The CIFAR-10 dataset. <http://www.cs.toronto.edu/~kriz/cifar.html>.