Received: August 11, 2023. Revised: September 16, 2023. 650

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Enhancement of Tree-Structured Deep Learning Classification Through Semantic Enabled Frequency Aware Data Augmentation

Nirmala Velusamy[1]* **Jayanthi Boopathy[1]**

[1]*School of Computer Studies, Rathnavel Subramanian College of Arts and Science, Coimbatore - 641402, Tamilnadu, India*
* Corresponding author's Email: nirmalavelusamy2018@gmail.com

Abstract: Nowadays, tree-structured deep learning classifier models have been widely used in different applications to ensure effective feature representation and learning. Amongst, dimensional sentiment analysis is the most interactive research field, which intends to identify continuous numerical values in the valence-arousal (VA) space. To achieve this, a tree-structured regional convolutional neural network with long short-term memory (T-CNN-LSTM) model was developed, which predicts the VA ratings of the texts for sentiment analysis. In contrast, the effect of a low prediction rate and difficulty of feature learning in a small number of class samples was not analyzed. Hence, this manuscript proposes an adversarial T-CNN-LSTM (A-T-CNN-LSTM) model for predicting the VA to achieve more fine-grained sentiment analysis. This model develops a semantic-enabled frequency-aware generative adversarial network (SFGAN) to produce more adversarial samples using the generator network and decrease the spectral data loss of the discriminator. It embeds the frequency-aware categorizer (FAC) into the discriminator to determine the input veracity in the spatial and spectral domains. Besides, semantic restricted sampling is employed in SFGAN for synthesizing the image subject to a semantic mask. Further, the created samples are classified by the T-CNN-LSTM for predicting the VA scores of sentences. Finally, the experimental results exhibit that the A-T-CNN-LSTM on stanford sentiment Treebank (SST) and CIFAR-10 databases achieves 90.12% and 91% accuracy than the other tree-structured CNNs.

Keywords: Dimensional sentiment analysis, Tree-structured deep learning, Valence arousal prediction, T-CNN-LSTM, GAN, Semantic restricted sampling.

1. Introduction

Since deep neural models have gained popularity in a variety of applications, tree-based techniques like decision trees (DTs) and random forests (RFs) are usually the dominant premise class in learning difficulties requiring metadata. These techniques are typically the successful strategy in Kaggle problems since they have many notable advantages [1]: they can deal with a wide range of attribute classes, they are unaffected by data quantity and they conduct a simple kind of attribute extraction by comparing connectives of result trees. Such qualities are essential in the superiority of tree-based techniques over original information.

Deep learning architectures such as DNN, CNN, recurrent neural network (RNN), and others have emerged as apparent choices when the input possesses a positional proximity characteristic (viz. text and multimedia data) [2]. In certain cases, such as image classification, by constraining the paradigm to use previous knowledge of spatial properties (e.g., interpretation and dimension invariance), these architectures may generate task-sensitive hypotheses that can completely remove the need for domain specialists. However, designing DNNs that operate with the tree-based techniques in the scenario of original data is sometimes incredibly challenging [3-5]. The classic fully connected networks (FCNs) that lack any inductive bias toward raw high-dimensional data are generally unfair to tree-based techniques for raw data [6].

Some research is being conducted to develop NNs for raw data. Most of these systems rely on classic DT learning in their loops and there is still no

widely accepted neural design that can successfully adapt tree-based approaches. This issue makes it impossible or complicated to apply neural designs in several contexts and it highlights a gap in the knowledge of DNNs.

To solve this problem, CNNs have emerged as the preferred architecture for large-scale image categorization in recent years. With the greater availability of massive volumes of labeled training data, supervised training has emerged as the fundamental concept for training CNNs to classify images [7]. The CNN is typically learned on a database comprising a vast number of annotated images. The network is learned to retrieve important characteristics from images and categorize them. This learned framework is then applied to identify specific unlabelled images [8]. During learning, every training sample is given to the network simultaneously. Nevertheless, each of these facts is not obtained concurrently nowadays and information is rather collected progressively over a period. This necessitates the development of frameworks capable of learning new data when it becomes accessible.

The CNN integrates feature extraction and classification into a single coherent structure within that framework. Updating one segment of the feature space has an instantaneous impact on the entire framework. The other concern with iteratively learning the CNN is catastrophic forgetting. If a learned CNN is reprogrammed only on fresh data, prior characteristics learned from historical information are discarded. This requires that historical information be used while updating fresh data. To combat the catastrophic forgetting problem and handle the characteristics trained in the prior process, an adaptive hierarchical network design called tree-CNN (T-CNN) [9].

The T-CNN network is composed of CNNs that develop hierarchically because new labels are adopted. This network includes new labels such as new leaves to the hierarchical design. The branching depends on the similarity of features between new and previous labels. The initial nodes of this framework allocate the input into coarse super-classes and better categorization is achieved because of approaching the leaves of the network. This framework facilitates leveraging the convolution layers trained before being utilized in the new larger network. But it was not able to learn the features extracted from the task-relevant sentences. From this perspective, the T-CNN-LSTM [10] was designed to execute more fine-grained sentiment analysis. This model predicted the VA scores of sentences by the regional CNN and LSTM. First, a portion of the sentence was taken as a region by the regional CNN that splits the given sentence into many regions to extract essential emotional data. The extracted data is weighted based on their influence on the prediction of VA. Then, the data from each region is combined by the LSTM for predicting VA. By merging these two networks, local data within texts and long-range correlations among texts are taken in the prediction task. Later, a region partition mechanism is applied to find task-related expressions and clauses to integrate organized data into the prediction of VA. But it does not analyze the impact of a low prediction rate and difficulty of feature learning in a small number of class varieties.

Thus, the A-T-CNN-LSTM model is developed in this article to improve the VA prediction efficiency for sentiment analysis. In this model, a GAN is adopted to create the adversarial samples for the small number of classes in the database. However, the problem of high-frequencies loss in the discriminator of a standard GAN is not solved and the generator has insufficient reward received from the discriminator to train high-frequency data features, yielding considerable spectrum discrepancies between created and actual images. To solve this problem, the SFGAN is developed. This SFGAN is a modification of GAN, which reduces the loss of spectral data in the discriminator. In this SFGAN, the FAC is embedded into the discriminator to estimate the input's veracity in the spatial and spectral domains. Also, a semantic restricted sampling is introduced in SFGAN to synthesize a picture related to a semantic mask. This process is achieved by formulating an optimization dilemma, which obtains the accurate latent vector for the GAN's picture creation when concerning the person's requirement. Thus, the SFGAN generates more samples to solve the data imbalance issue and predicts the VA using T-CNN-LSTM effectively.

The rest of this paper is organized as follows: Section 2 covers a literature survey regarding the tree-based deep learning frameworks in different applications. Section 3 describes the A-T-CNN-LSTM model, whilst section 4 demonstrates its performance. Section 5 abridges the entire article and gives upcoming works.

2. Literature review

The semantic visual concepts were presented [11] to analyze CNN predictions quantitatively and semantically. Also, a method was presented to train the DT without robust supervision for interpretations. In this method, the feature interpretations in high convolution layers were decomposed by the DT into elementary concepts of object parts for prediction. But it needs more training samples to increase the

accuracy.

A boosting cascade deep forest (BCDForest) classifier framework [12] was developed to categorize the tumor subcategories from the gene expression data. A multi-class-grained analysis was adopted to support the variety of ensembles via various learning information. The forest's characteristics for training were considered based on the sliding window. The out-of-bagging was utilized to measure the network loss and allocate a self-reliance weight to all forests to get the proper outcomes. A variation-based method was developed to enrich significant characteristics in training forests at all levels of a cascade forest. But its robustness was less while using the high-dimensionality small-scale and class-imbalanced datasets.

The TreeUNet framework [13] was designed using a dynamic method to improve pixel-level categorization efficiency. The T-CNN module with all nodes denoting a ResNext block was built dynamically according to the deep semantic framework structure. The T-CNN module integrates multiscale attributes and trains the framework's optimal weights while transmitting attribute mappings through merging connections. But the training samples were not adequate.

A deep fuzzy tree (DFT) framework [14] was developed to resolve the large-scale hierarchical visual categorization with more classes via substituting the softmax function in the deep learner. Also, a novel dual fuzzy inter-class correlation measure was introduced to configure the tree learning and base classifiers. But the accuracy was degraded when the node number and tree depth were not chosen appropriately.

A novel technique was presented to improve the usability of rule extraction for deep learners [15]. Initially, the CNN was decomposed into a feature extractor and a classifier. After that, the DT was extracted only from the classifier and various partitioned labeled images were leveraged to train the concepts of each feature. Moreover, the human-readable DTs were extracted from the CNNs to construct CNN2DT and allow users to find the surrogate DTs. But the DT structures were vaguely unstable while using more training images.

A tag-guided hyper-recursive neural network (TG-HRecNN) [16] was designed that adopts hyper-networks into RecNNs to consider as part-of-speech (POS) labels of terms and create the lexical variables with dynamism. Also, the data merging unit was designed to integrate POS labels and lexical data at all nodes to direct the compilation task. But it cannot process high-level tasks since it needs encoded sentence embedding.

The efficacy of using a deep learning-based DT model was analyzed to detect COVID-19 [17] in lung X-ray images. Three binary DTs were used in this model, which was trained by the CNN using the PyTorch package. The main DT can classify the X-ray images of the lungs as normal or atypical. The second and third DTs can reveal unusual scans including TB and COVID-19 signs, respectively. But the outcomes were not reliable without using pathologically verified data and the number of training samples was not sufficient. An effective semantic partition of satellite images was presented [18]. Initially, artifacts were removed from the satellite images and the semantic representation based on T-CNN was applied to capture the semantic areas of downscaled images. Further, the deep forest classifier was utilized to find the possible areas. However, the training samples were not adequate.

The new deep DT (DDT) classifier model [19] was designed to recognize the cyberbullying tests from the Twitter engine inside the smart city. In this model, the hidden layers of DNN were utilized as its tree node to analyze the input elements. But it was appropriate for small-scale databases. A tree-RNN framework [20] was developed to classify the network traffic. A binary tree was used to ensure that each classifier in the tree structure implements the small classification and the specific split rules were provided for the number of traffic classes. The RNN model was used to train the time-related characteristics of the data and the cosine similarity was used to estimate which classes were qualified to a similar node. But the accuracy was not high due to the imbalanced databases.

A new model depending on CNN and XGBoost called ConvXGB [21] was designed to solve classification issues. It comprised many stacked convolutional layers to learn the input features, followed by the XGBoost in the final layer to predict the labels. But its accuracy was less since it cannot learn temporal correlation among given data.

The RF-CNN with features (RF-CNN-F) [22] was developed to diagnose coronary artery disease depending on cardiac magnetic resonance. The high-dimension images were transformed into low-dimension ones, which were then given to the CNNs to capture the important features automatically. Such features were further used to create the DTs for classifying coronary artery disease based on the majority voting scheme. But the RF performance was influenced by the number of features utilized in the DTs. An attention-driven tree-structured convolution LSTM (ADT-ConvLSTM) model [23] was developed for high-dimensional data modeling. But its computational cost was high.

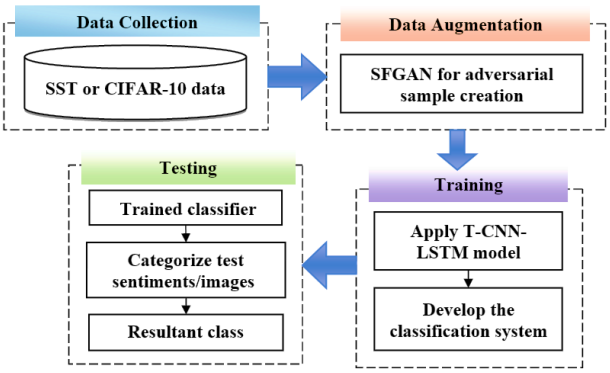

Figure. 1 Schematic representation of presented study

Table 1. Lists of notations

Notations	Description
G	Generator network
D	Discriminator network
N	No. of layers in generator
m	No. of semantic categories
T_i	Linear conversion matrix
x_i	Feature maps in layer i
S	Semantic partition
x	Input image/text
z	Latent vector
$u_i^{\uparrow}$	Upsampling output
$\mathcal{L}_{enp}$	Cross-entropy loss function
Y_{ij}	Semantic category at pixel (i,j)
$S_{ij}[k]$	Unregularized logarithmic possibility for k^{th} semantic category
$\mathbb{S}$	Batch size
C	Spectral classifier
$f(m,n)$	Discrete 2D signal
$\mathcal{F}$	Discrete Fourier transform
k,l	Spectral coordinates
θ	Polar coordinate
v	Reduced spectral interpretation vector
r	Radial distance
$\mathcal{L}_{spectral}$	Spectral categorization loss
$C(\phi(x))$	Spectral veracity of x
P_g	Generator's distribution
D^{SS}	Enhanced discriminator
λ	Hyperparameter
$\mathcal{L}_{advl}$	Adversarial loss

In [24], an optimized hierarchical T-CNN model with sheep flock optimization was developed to predict workload and increase power efficiency in cloud computing. But the complexity of this model was high and it was a non-convex problem.

3. Proposed methodology

This section briefly describes the A-T-CNN-LSTM model. A schematic representation of the presented study is illustrated in Fig. 1. At first, the different open-source databases are collected and then the proposed SFGAN is applied to augment the training samples by creating more adversarial samples. The created and actual samples are further trained by the T-CNN-LSTM classifier. Moreover, the trained classifier is used to classify the test data into positive and negative or test images into different classes. Table 1 lists the notations used in this study.

3.1 Design of SFGAN

Typically, the GAN comprises learning the generator (G) and discriminator (D) networks, where G trains to regenerate the target data distribution. But, it is complex to unambiguously define the semantics of created data samples. So, this SFGAN aims to efficiently define the GANs' semantic interpretation so it allows semantic controller in G. In this network, a linear feature map conversion is performed to capture the created image semantics. Compared with the GAN's nonlinear image creation task, the easiest linear conversion has an understandable geometric representation. Also, it applies semantic restricted sampling to handle the image creation by producing images regarding a person's requirement of the target semantic pattern.

3.1.1. Enhanced generator

Construct a generator network (G) model comprising N layers and creating images with m semantic categories. It aims to find the potential correlation between its feature maps and output image semantics. So, a linear conversion matrix (T_i) is employed on all feature maps x_i to estimate a semantic map of the layer i. By adding each map, a semantic partition (S) of the GAN's resultant image is predicted. The structure of an enhanced generator in the SFGAN model is shown in Fig. 2, which describes how synthesizing an input image/text (x) from a latent vector z, G creates a sequence of internal x_i.

To obtain an effective GAN model, the feature maps $\{x_i\}_{i=1}^{N-1}$ are decoded to retrieve the resultant image/text's semantic partition S. This S is a linear conversion of each $x_i$and is described as:

$$S = \sum_{i=1}^{N-1} u_i^{\uparrow}(T_i \cdot x_i) \quad (1)$$

In Eq. (1), $T_i \in \mathbb{R}^{m \times c_i}$ transforms $x_i \in \mathbb{R}^{c_i \times w_i \times h_i}$ into a semantic map $T_i \cdot x_i \in \mathbb{R}^{m \times w_i \times h_i}$ using a tensor reduction along the depth axis. After that, the output from all layers is upsampled (represented as $u_i^{\uparrow}$) to the resultant image resolution.

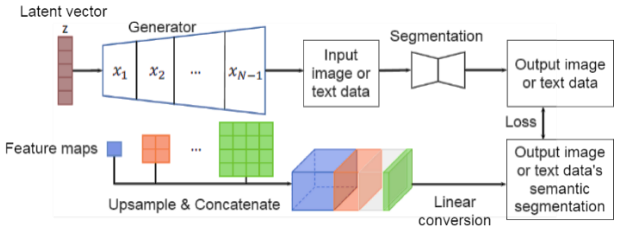

Figure. 2 Enhanced generator in SFGAN model

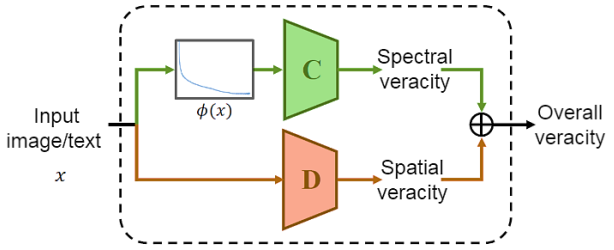

Figure. 3 Enhanced discriminator in SFGAN model

The summation ranges over each internal layer, exclusive of the final layer N that yields the resultant image. The output $S \in \mathbb{R}^{m \times w \times h}$ contains a similar spatial resolution $w \times h$ as the resultant image. All pixels S_{ij} is a $m \times 1$ vector, defining the pixel's unregularized logarithmic possibilities and denoting every m semantic class. This strategy is known as a linear semantic extractor.

The learning task of this strategy is supervised via pixel-level labeling of semantics. But, it is unfeasible to physically label the huge amount of created images. So, off-the-shelf pre-trained segmentation frameworks are used for semantic labeling. To learn this mechanism and construct a well-learned SFGAN model, its latent space is randomly sampled to create a collection $\mathbb{S}$ of created images. While creating all images in $\mathbb{S}$, the model's feature maps $\{x_i\}_{i=1}^{N-1}$ are also captured. Such x_i are linearly converted by Eq. (1) to estimate the images' semantic mask, which is evaluated by the output from the pre-trained semantic segmentation model to determine the classical cross-entropy loss value as:

$$\mathcal{L}_{enp} = \frac{1}{w \cdot h} \sum_{\substack{1 \le i \le w \\ 1 \le j \le h}} \left[\log \left(\sum_{k=1}^{m} e^{(S_{ij}[k])} \right) S_{ij}[Y_{ij}] \right] \quad (2)$$

In Eq. (2), Y_{ij} indicates the semantic category at pixel (i, j) and $S_{ij}[k]$ indicates the respective unregularized logarithmic possibility for k^{th} semantic category estimated using the linear semantic extractor. Moreover, T_i are adjusted by reducing the estimated loss (determined by considering the mean loss over image batches in $\mathbb{S}$). Accordingly, G of the SFGAN can create adversarial samples and extract their semantics.

3.1.2. Enhanced discriminator

Initially, a spectral classifier C is introduced to identify frequency spectrum divergence between actual and created images. After that, C is integrated into D of GANs to improve its capability in the spectral domain and minimize the spectrum divergence. Fig. 3 shows the structure of an enhanced discriminator in the SFGAN model.

To solve the problem of high-frequency loss in D, a simple method is to distinguish in the frequency domain instead of the spatial domain. For a discrete 2D signal $f(m, n)$ defining an image of dimension $M \times N$, its discrete fourier transform $\mathcal{F}$ is calculated in Eq. (3):

$$\mathcal{F}(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-2\pi i \frac{km}{M}} e^{-2\pi i \frac{ln}{N}} \quad (3)$$

For the spectral coordinates $k = 0, \ldots, M-1$ and $l = 0, \ldots, N-1$. After that, it is converted from Cartesian coordinates k and l to polar coordinates r and θ to signify the frequencies of various spectrums in Eq. (4),

$$\mathcal{F}(r, \theta) = \mathcal{F}(k, l) : r = \sqrt{k^2 + l^2}, \theta = atan2(l, k) \quad (4)$$

The reduced spectral interpretation v is obtained by azimuthally averaging over θ as:

$$v(r) = \frac{1}{2\pi} \int_0^{2\pi} |\mathcal{F}(r, \theta)| d\theta \quad (5)$$

Eq. (5) defines the average intensity of the signal relating to the radial distance r. The shrunk spectral interpretation can smooth the spectrum variations at high frequencies. For a given image x, the grayscale element is utilized to obtain its spectral vector v and the function is defined as $v = \phi(x)$. The spectral categorization loss is defined as:

$$\mathcal{L}_{spectral} = \mathbb{E}_{x \sim P_{data}(x)} \left[logC\left(\phi(x)\right) \right] + \mathbb{E}_{x \sim P_g(x)} \left[log\left(1 - C\left(\phi(x)\right)\right) \right] \quad (6)$$

In Eq. (6), $C\left(\phi(x)\right)$ determines the spectral veracity of x and P_g is G's distribution. Because x is realistic when it is accurate in the spatial and frequency domains, the veracity of x is determined with the mixture of spatial and spectral veracity.

In this SFGAN, C is integrated into D of GANs to support G, which learns the high-frequency and semantics of the image. This enhanced discriminator (D^{SS}) has 2 units: a vanilla discriminator (D) that estimates the spatial veracity and C. So, D^{SS} is used

to determine the input veracity in the spatial and spectral domain. Also, the complete veracity of x is denoted by

$$D^{SS}(x) = \lambda D(x) + (1-\lambda)C\big(\phi(x)\big) \quad (7)$$

In Eq. (7), λ denotes the hyperparameter, which regulates the virtual significance of the spatial and spectral veracity. The models' adversarial loss is represented by

$$\mathcal{L}_{advl} = \mathbb{E}_{x\sim P_{data}(x)}[\log D^{SS}(x)] + \mathbb{E}_{x\sim P_g(x)}\big[log\big(1-D^{SS}(x)\big)\big] \quad (8)$$

In Eq. (8), P_g is G's distribution. To train this D network, C, D and G are fine-tuned using the below gradients in Eqs. (9a) to (9c).

$$\theta_c \leftarrow -\nabla_{\theta_c}\mathcal{L}_{spectral} \quad (9a)$$

$$\theta_d \leftarrow -\nabla_{\theta_d}\mathcal{L}_{advl} \quad (9b)$$

$$\theta_g \leftarrow -\nabla_{\theta_g}\mathcal{L}_{enp} \quad (9c)$$

Because less information of x is removed in the spectral vector $\phi(x)$, it is observed that it is not able to give a useful gradient for adversarial learning, which influences the model efficiency. Therefore, consider that the backpropagation procedure of Eq. (8) didn't pass via C and $C\big(\phi(x)\big)$ acts as a spectral modulating parameter to $\mathcal{L}_{advl}$. Moreover, the gradients of D and G are given in Eqs. (10) & (11):

$$\theta_d \leftarrow \frac{1}{1-D(x)+\frac{1-\lambda}{\lambda}\big(1-C(\phi(x))\big)}\nabla_{\theta_d}D(x) \quad (10)$$

$$\theta_g \leftarrow -\frac{1}{D(x)+\frac{1-\lambda}{\lambda}C(\phi(x))}\nabla_x D(x)J_{\theta_g}G(z) \quad (11)$$

From the abovementioned gradients, it is noticed that this SFGAN executes a hard sample extraction, wherein hard is represented in the frequency domain. For D, if $C\big(\phi(x)\big) \to 1$, the created image x contains high-quality spectral features and is a hard sample to be categorized as bogus. For G, x refers to a hard sample, if $C\big(\phi(x)\big) \to 0$. This defines that x contains ineffective spectral veracity and requires further consideration from G.

If x refers to a hard sample in the frequency domain, the gradients of D and G are up-weighted, which encourages the model to train the spectral distribution of the actual image. Thus, this SFGAN can generate more adversarial data/images similar to the given input data/images for effective training of the T-CNN-LSTM model, which classifies the text's sentiments or images appropriately.

4. Experimental result

This section presents the A-T-CNN-LSTM's effectiveness by executing it in Python 3.7.8 using 2 different databases: SST and CIFAR-10. In the SST database [24], there are 8,544 learning texts, 2,210 test texts and 1,101 validation texts. All texts are graded on a particular axis (valence) that ranges from 0 to 1. In contrast, the CIFAR-10 database [25] contains 60000 color images of dimension 32×32 in 10 different categories, with 6000 images per category. Of these, 50000 images are used for learning and 10000 are used for testing. Each of the 10000 images is divided into 5 learning batches and 1 test batch. The test batch has 1000 images from all categories, picked randomly. The residual images are used in learning batches arbitrarily, but a few batches might comprise several images from a certain category compared to another. Amongst, the learning batches have 5000 images from all categories. To measure the performance of the proposed model, existing models including T-CNN-LSTM [10], T-CNN [9], Tree-RNN [20], ConvXGB [21], RF-CNN-F [22], and ADT-ConvLSTM [23] are also implemented and tested using the above-considered datasets. Table 2 lists parameter settings for the proposed A-T-CNN-LSTM and existing models.

The metrics used for comparison analysis are defined as follows:

- **Accuracy:** It defines a proper categorization of text's sentiment or image class among the overall samples tested.

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

TP is the number of +ve texts that are correctly classified as +ve, TN is the number of +ve texts that are correctly classified as –ve, FP is the number of –ve texts that are incorrectly classified as +ve and FN is the number of +ve texts that are incorrectly classified as –ve.

- **Precision:** It is determined by

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

- **Recall:** It is calculated by

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

Table 2. Parameter settings for existing and proposed A-T-CNN-LSTM model

Framework	Parameters	Range
ConvXGB [21]	No. of convolutional layer	2
	Output depth of convolutional layer	4
	Filter size	2×
	Stride of the filter	1
	No. of trees	10
	Tree depth	4
	No. of epoch	50
	Learning rate	0.001
RF-CNN-F [22]	No. of convolutional layer	2
	No. of fully connected layer	1
	No. of filters in each convolutional layer	32
	Kernel size	3×3
	Stride size	2
	Activation function for hidden layers	ReLU
	L_2-regularization coefficient	0.001
	Batch size	256
	No. of epochs	20
	Learning rate	0.001
ADT-ConvLSTM [23]	No. of LSTM units	64
	LSTM activation function	tanh
	No. of filters	32
	Kernel size	3×3
	Stride size	1
	Learning rate	0.001
	Batch size	32
	No. of epochs	100
	Dropout	0.2
T-CNN [9]	Learning rate	0.1
	Batch size	64
	Epoch	200
	Weight decay	0.001
	Momentum	0.9
	Dropout	0.2
Tree-RNN [20]	Learning rate	0.001
	Batch size	64
	Epochs	100
	No. of hidden states	120
T-CNN-LSTM [10] and proposed A-T-CNN-LSTM	No. of filters	60
	Filters length	3
	Pool length	2
	No. of hidden states	120
	Optimizer	Adam
	Learning rate	0.0001
	Batch size	32
	No. of epochs	50
	Dropout	0.25

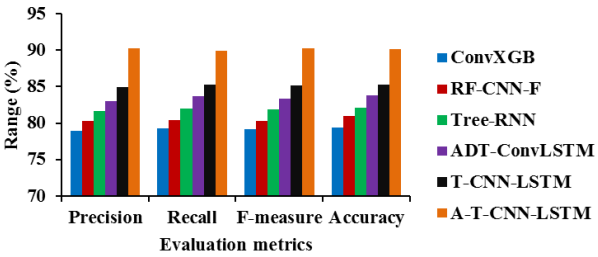

Figure. 4 Performance analysis of different classification models on SST database

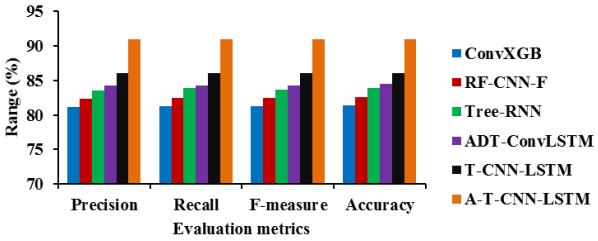

Figure. 5 Performance analysis of different classification models on CIFAR-10 database

- **F-measure:** It is determined by

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

Fig. 4 demonstrates the performance of different models applied on the SST database to classify the text's sentiments, which clarifies the A-T-CNN-LSTM attains better precision, recall, f-measure and accuracy compared to the other classifier models because of learning more adversarial samples generated by the SFGAN. The precision values obtained by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 78.96%, 80.29%, 81.62%, 83.05%, 84.93% and 90.31%, correspondingly. The recall values determined by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 79.23%, 80.35%, 82.04%, 83.68%, 85.31% and 89.93%, respectively. The f-measure obtained by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 79.1%, 80.32%, 81.83%, 83.37%, 85.31% and 90.12%, respectively.

Also, the accuracy of the A-T-CNN-LSTM model is 13.54%, 11.34%, 9.76%, 7.61% and 5.64% greater than the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, and T-CNN-LSTM for classifying the test's sentiments effectively.

In Fig. 5, the performance of different models tested on the CIFAR-10 database is shown, which signifies that the A-T-CNN-LSTM model accomplishes better efficiency compared to the other

models due to the extracting of image semantics and learning more adversarial samples generated by the SFGAN. The precision values obtained by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 81.18%, 82.36%, 83.51%, 84.25%, 86% and 91%, correspondingly. The recall values determined by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 81.27%, 82.52%, 83.86%, 84.31%, 86% and 91%, respectively. The f-measure obtained by the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, T-CNN-LSTM and A-T-CNN-LSTM are 81.23%, 82.44%, 83.69%, 84.5%, 86% and 91%, respectively. Also, the accuracy of the A-T-CNN-LSTM model is 11.88%, 10.17%, 8.46%, 7.69% and 5.81% greater than the ConvXGB, RF-CNN-F, Tree-RNN, AT-ConvLSTM, and T-CNN-LSTM for classifying the test's sentiments effectively.

5. Conclusion

This paper developed the A-T-CNN-LSTM model to estimate the VA and analyze the sentiments from images/texts. In this model, the SFGAN was designed, which produces the number of adversarial samples by the generator network. This generator network can apply linear transformation and semantic restricted sampling to facilitate semantic control during the creation phase. By using this new generator, the created image semantics were captured by the linear transformation of feature maps. The FAC was embedded into the discriminator unit to evaluate the input veracity in the spatial and spectral spaces. Thus, this SFGAN can produce more adversarial samples, which were then classified by the T-CNN-LSTM model for sentiment analysis. Finally, the experimental results revealed that the A-T-CNN-LSTM on SST and CIFAR-10 databases has an accuracy of 90.12% and 91%, respectively compared to the other models for sentiment analysis/image recognition.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, validation, Nirmala; formal analysis, investigation, Jayanthi; resources, data curation, writing—original draft preparation, Nirmala; writing—review and editing, Nirmala; visualization; supervision, Jayanthi.

References

[1] E. K. Ampomah, Z. Qin, and G. Nyame, "Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement", *Information*, Vol. 11, No. 6, pp. 1-21, 2020.

[2] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions", *SN Computer Science*, Vol. 2, No. 6, pp. 1-20, 2021.

[3] Y. Yang, I. G. Morillo, and T. M. Hospedales, "Deep Neural Decision Trees", *ArXiv Preprint ArXiv:1806.06988*, 2018.

[4] N. Frosst and G. Hinton, "Distilling a Neural Network into a Soft Decision Tree", *ArXiv Preprint ArXiv:1711.09784*, 2017.

[5] H. Zhu, X. Li, P. Zhang, G. Li, J. He, H. Li, and K. Gai, "Learning Tree-Based Deep Model for Recommender Systems", In: *Proc. of the 24th ACM International Conf. on Knowledge Discovery & Data Mining*, pp. 1079-1088, 2018.

[6] S. Abpeikar, M. Ghatee, G. L. Foresti, and C. Micheloni, "Adaptive Neural Tree Exploiting Expert Nodes to Classify High-Dimensional Data", *Neural Networks*, Vol. 124, pp. 20-38, 2020.

[7] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", *Neural Computation*, Vol. 29, No. 9, pp. 2352-2449, 2017.

[8] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", *Artificial Intelligence Review*, Vol. 53, No. 8, pp. 5455-5516, 2020.

[9] D. Roy, P. Panda, and K. Roy, "Tree-CNN: A Hierarchical Deep Convolutional Neural Network for Incremental Learning", *Neural Networks*, Vol. 121, pp. 148-160, 2020.

[10] J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 581-591, 2019.

[11] Y. Liu, Y. Dou, R. Jin, and P. Qiao, "Visual Tree Convolutional Neural Network in Image Classification", In: *Proc. of IEEE 24th International Conf. on Pattern Recognition*, pp. 758-763, 2018.

[12] Y. Guo, S. Liu, Z. Li, and X. Shang, "BCDForest: A Boosting Cascade Deep Forest Model towards the Classification of Cancer Subtypes Based on Gene Expression Data",

BMC Bioinformatics, Vol. 19, No. 5, pp. 1-13, 2018.

[13] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "TreeUNet: Adaptive Tree Convolutional Neural Networks for Subdecimeter Aerial Image Segmentation", *Journal of Photogrammetry and Remote Sensing*, Vol. 156, pp. 1-13, 2019.

[14] Y. Wang, Q. Hu, P. Zhu, L. Li, B. Lu, J. M. Garibaldi, and X. Li, "Deep Fuzzy Tree for Large-Scale Hierarchical Visual Classification", *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 7, pp. 1395-1406, 2019.

[15] S. Jia, P. Lin, Z. Li, J. Zhang, and S. Liu, "Visualizing Surrogate Decision Trees of Convolutional Neural Networks", *Journal of Visualization*, Vol. 23, No. 1, pp. 141-156, 2020.

[16] G. Shen, Z. H. Deng, T. Huang, and X. Chen, "Learning to Compose Over Tree Structures via POS Tags for Sentence Representation", *Expert Systems with Applications*, Vol. 141, pp. 1-8, 2020.

[17] S. H. Yoo, H. Geng, T. L. Chiu, S. K. Yu, D. C. Cho, J. Heo, and H. Lee, "Deep Learning-Based Decision-Tree Classifier for COVID-19 Diagnosis from Chest X-ray Imaging", *Frontiers in Medicine*, Vol. 7, pp. 1-8, 2020.

[18] Y. H. Robinson, S. Vimal, M. Khari, F. C. L. Hernández, and R. G. Crespo, "Tree-Based Convolutional Neural Networks for Object Classification in Segmented Satellite Images", *The International Journal of High Performance Computing Applications*, pp. 1-14, 2020.

[19] N. Yuvaraj, V. Chang, B. Gobinathan, A. Pinagapani, S. Kannan, G. Dhiman, and A. R. Rajan, "Automatic Detection of Cyberbullying Using Multi-Feature Based Artificial Intelligence with Deep Decision Tree Classification", *Computers & Electrical Engineering*, Vol. 92, pp. 1-13, 2021.

[20] X. Ren, H. Gu, and W. Wei, "Tree-RNN: Tree Structural Recurrent Neural Network for Network Traffic Classification", *Expert Systems with Applications*, Vol. 167, pp. 1-9, 2021.

[21] S. Thongsuwan, S. Jaiyen, A. Padcharoen, and P. Agarwal, "ConvXGB: A New Deep Learning Model for Classification Problems Based on CNN and XGBoost", *Nuclear Engineering and Technology*, Vol. 53, No. 2, pp. 522-531, 2021.

[22] F. Khozeimeh, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi, R. Alizadehsani, and S. M. S. Islam, "RF-CNN-F: Random Forest with Convolutional Neural Network Features for Coronary Artery Disease Diagnosis Based on Cardiac Magnetic Resonance", *Scientific Reports*, Vol. 12, No. 1, p. 11178, 2022.

[23] Y. Lu, B. Kong, F. Gao, K. Cao, S. Lyu, S. Zhang, and X. Wang, "Attention-Driven Tree-Structured Convolutional LSTM for High Dimensional Data Understanding", *Frontiers in Physics*, Vol. 11, p. 1095277, 2023.

[24] T. S. C. Chetty, V. Bolshev, S. S. Subramanian, T. Chakrabarti, P. Chakrabarti, V. Panchenko, and Y. Daus, "Optimized Hierarchical Tree Deep Convolutional Neural Network of a Tree-Based Workload Prediction Scheme for Enhancing Power Efficiency in Cloud Computing", *Energies*, Vol. 16, No. 6, p. 2900, 2023.

[25] https://nlp.stanford.edu/sentiment/treebank.html.

[26] http://www.cs.toronto.edu/~kriz/cifar.html.