Multi-class Emotion Classification Using EEG Signals

Divya Acharya[1(✉)], Riddhi Jain[2], Siba Smarak Panigrahi[3], Rahul Sahni[4], Siddhi Jain[4], Sanika Prashant Deshmukh[5], and Arpit Bhardwaj[1]

[1] Bennett University, Greater Noida, India
{da9642,arpit.bhardwaj}@bennett.edu.in
[2] Jaipur Engineering College and Research Centre, Jaipur, India
riddhi.ten.jain@gmail.com
[3] Indian Institute of Technology Kharagpur, Kharagpur, India
sibasmarak.p@iitkgp.ac.in
[4] Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Jalandhar, India
rsahni9785@gmail.com, siddhi.ten@gmail.com
[5] Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India
dsanika0@gmail.com

Abstract. Recently, the availability of large EEG datasets, advancements in Brain-Computer interface (BCI) systems and Machine Learning have led to the implementation of deep learning architectures, especially in the analysis of emotions using EEG signals. These signals can be generated by the user while performing various mental, emotional and physical tasks thus, reflecting the brain functionality. Extracting the important feature values from these unprocessed signals remain a vital step in the deployment. Fast Fourier Transformation proves to be better than the traditional feature extraction techniques. In this paper we have compared the deep learning models namely Long Short-term Memory (LSTM) and Convolutional Neural Network (CNN) on 80–20 and 75–25 Train-Test splits. The best result was obtained from LSTM classifier with an accuracy of 88.6% on the liking emotion. CNN also gave a good accuracy of 87.72% due to its capability to extract spatial feature from the input signals. Thus, both these models are quite beneficial in this context.

Keywords: Emotion recognition · EEG · Convolutional neural network · Long short-term memory networks · Deep learning

1 Introduction

Electroencephalography (EEG) signals track and record brain activities with small metal discs with thin wires (electrodes) placed on the scalp. Analysis of EEG signals help researchers, doctors to assess and diagnose brain and mental diseases. Due to complex nature, noise, artefacts in EEG signals, and data from many patients making the EEG signal analysis process time-consuming and may not always be accurate. Careful analysis of EEG with computer algorithms provides valuable insights and helps better

D. Garg et al. (Eds.): IACC 2020, CCIS 1367, pp. 474–491, 2021.
https://doi.org/10.1007/978-981-16-0401-0_38

understanding of diseases like Stroke Diagnosis, Epilepsy Diagnosis, Autism Diagnosis, Sleep Disorder Diagnosis, Dementia Diagnosis, Alcoholism Diagnosis, Anesthesia Monitoring, Coma and Brain Death, Brain Tumor Diagnosis [1]. The new and emerging Brain-Computer Interfaces (BCIs) technology captures EEG signals and analyses and translates it into the command to carry out the desirable tasks. Different researchers have used different Machine Learning algorithms for improvement in the study of EEG signals.

Emotions play a significant role in human beings as these are associated with the brain and bring neurophysiological changes in thoughts, feelings, behavioral responses, and a degree of pleasure or displeasure.

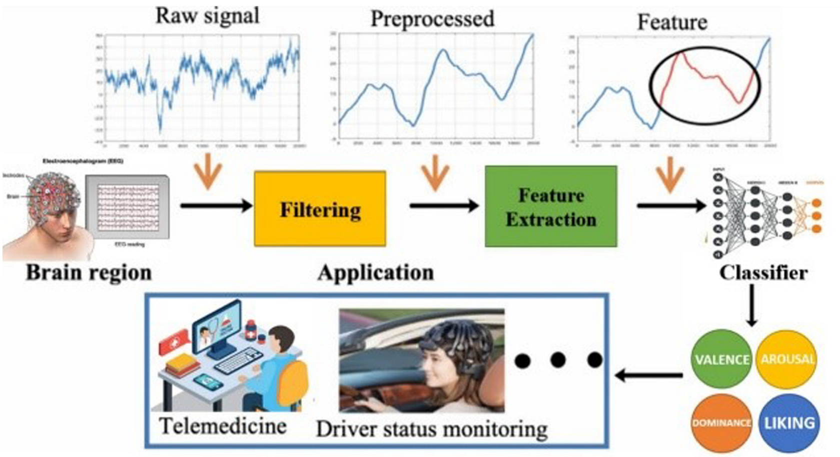

Fig. 1. Proposed model architecture for emotion recognition using EEG signal processing

Emotions are complex in nature as they involve various components like subjective experience, cognitive processes, psychophysiological changes, and instrumental and expressive behavior. Everyday interaction between humans like facial expression, voice, the text provides primary data to researchers to identify human emotion, but it may not be exactly how they are feeling but rather as they feel others would respond. EEG signals to aid in identifying emotions as it gives a better understanding of participant's underlying responses captured at the time of observation [2]. As shown in Fig. 1, signals are captured using electrodes. These signals are then filtered by removing the noises and artifacts, which can be done by bandpass filtering. Removal of artifacts entirely is not possible, as removal of artifacts entirely may result in the loss of some valuable information. After filtering data, necessary features are extracted and shaped correctly to fit into a classifying model for the analysis of several emotions. Once the model is implemented successfully, it can be used to deploy various applications. Therefore, it offers high accuracy for recognition of emotions as compared to voice or facial expression. Various researchers worked on EEG data to classify emotional states using machine learning algorithms like random forest [3], Naive Bayes [4], KNN [5], SVM with RBF kernel function [6]. But very few researchers used deep learning algorithms like CNN [7] and RNN [8] to analyze EEG signals.

In this paper, we are proposing a distinct comparison between two deep learning architecture LSTM (Long Short-term Memory) & CNN (Convolution Neural Network), in which we have split the pre-processed DEAP data into two splits, i.e., 75–25 and 80–20 where we received a good result. We are using LSTM as sequences tasks as they can capture more dependency and predicts the sequence of data. The data used in this paper is not continuous but is sequential, so LSTM offers excellent results. For CNN, the prediction is much faster and is done in computationally efficient manner, that is why CNN is in this research paper.

2 Related Work

S Tripathy et al. (2017) [9] explored Deep Neural Network (DNN) and Convolutional Neural Network (CNN) for emotion classification on DEAP dataset. The proposed architecture of their DNN model is an input layer of 4040 units followed by 5000, 500 and 1000 hidden units in three hidden layers. The output layer is a 2 or 3 class softmax (Dunne and Campbell (1997) [10]) classifier depending upon the requirement. Further the proposed architecture of CNN was two convolutional layers, followed by Maxpooling and Dropout layers, which connects to Fully Connected layers to provide the output. They achieve an accuracy of 75.78% and 73.125% for DNN and 81.406% and 73.36% for CNN in 2 class (high and low) valence and arousal classification respectively. For 3 class classification (high, normal and low) of valence and arousal, the accuracy achieved is 58.44% and 55.70% for DNN and 66.79% and 57.58% for CNN.

W. Liu et al. (2016) [11] extract features by the Bimodal Deep Auto-encoder (BDAE). They design two Restricted Boltzmann Machine (RBM), one for EEG (EEG RBM) and other for eye movement features (eye RBM). They concatenate the hidden layers and obtain an upper RBM. The BDAE network is used for feature selection and they train linear SVM classifier on the high-level features extracted. The mean accuracies achieved with the BDAE network are respectively 91.01% and 83.25% on SEED and DEAP datasets.

S. Alhagry et al. (2017) [12] proposed 2-layer stacked LSTM architecture for emotion recognition on DEAP dataset. The first LSTM layer consists of 64 units with ReLU activation function, followed by a dropout layer with 0.2 probability. Second layer consists of 32 neurons with sigmoid activation function connected finally to a dense layer again with sigmoid activation. They divide valence, arousal, and liking to high/low class and respectively obtain an average accuracy of 85.65%, 85.45%, and 87.99% on DEAP dataset.

J. Zhang et al. (2016) [13] obtain average classification accuracy of 81.21% and 81.26% on valence and arousal respectively. They use Probabilistic Neural Networks (PNNs), which consists of four layers including the input layer, and the output layer. The second layer is termed as Pattern layer and the third layer is termed as Summation layer.

P. Zhong et al. (2020) [14] propose a Regularized Graph Neural Network (RGNN) for EEG based emotion recognition, in addition to two regularizers to make their model robust, node-wise domain adversarial training (NodeDAT) and emotion-aware distribution learning (EmotionDL). They beat the state-of-the-art results of bi-hemispheric discrepancy model (BiHDM) (Y.Li et al. (2019) [15]) with the average accuracies of 94.24%

and 79.37% for subject-dependent classification accuracy on SEED and SEED-IV (all-bands), while the BiHDM achieved average accuracies of 93.12% and 74.35% respectively. Further in case of subject-independent classification, RGNN obtained 85.30% and 73.84% mean accuracy respectively.

D. Acharya et al. (2020) [16] provide LSTM architecture for negative emotion classification and also briefly examines the human behavior in different age groups and gender. Their LSTM model, for four class negative emotion classification obtains classification accuracy of 81.63%, 84.64%, 89.73%, and 92.84% for 50–50, 60–40, 70–30 split of data, and 10-fold cross-validation. The models have been evaluated on both DEAP and SEED datasets.

A. Bhardwaj et al. (2014) [17] provide a novel Genetic Programming approach with hill-climbing integrated constructive crossover and mutation operators. They have estimated their classification accuracy to be 98.69%.

Another novel Genetic Programming approach with provision of a technique for hybrid crossover, intron deletion and mutation operation has been proposed in H. Bhardwaj et al. (2019) [18], which increases the accuracy of classification and also leads to a decrease in time complexity. This further suggests the possibility of a real-time Genetic Programming classifier for detection of epileptic seizures.

A new fitness function termed as Gap score (G score) has been proposed in D. Acharya et al. (2020) [19] to address imbalance in dataset. They propose a framework termed as GGP, a Genetic programming framework with G score fitness function. Their GGP framework provides 87.61% classification accuracy using EEG signals.

3 Methodology

In this section dataset description, feature extraction technique used, model architecture, and hyperparameter used for training the classifier including description of implementation tools are described next.

3.1 Dataset Description

The collection of the original DEAP dataset [20] was done in 2 parts. The first part was the online self-assessment where 14–16 subjects rated 120 YouTube music videos each of 1-min extract based on valence, arousal, dominance, likeness and familiarity all listed in online ratings.csv or.xls file. The second part involves physiological recordings and participant ratings of 32 volunteers. These unprocessed physiological clips were in BioSemi.bdf format. Out of 120 videos 40 were shown to each of them. Frontal face clips were also recorded for 22 subjects and could be downloaded from face_video.zip file. Apart from these, the dataset also had list and links of YouTube music videos and a participant questionnaire file containing all the answers given by each participant to the questions asked before the experiment. Original 512 Hz EEG signal were pre-processed to 128 Hz after down sampling, filtering, segmenting and removing all the artefacts like eyes blinking, muscle movements etc. They were present in MATLAB.mat and Python.dat format.

In each of these 32.dat files corresponding to each participant there were 2 arrays: Data and labels. Data was of 40 × 40 × 8064 dimensions. There were 40 channels in each video which in turn had 8064 EEG signal data that forms 322560 in total. The labels had 40 × 4 shape where 4 signifies valence, arousal, dominance and liking (Table 1). Python NumPy arrays is used and loaded.dat files using cPickle library and encoding latin1.

Table 1. Pre-processed dataset description

Data	40 × 40 × 8064 (video × channel × data)
Labels	40 × 4 video × label (Valence, Arousal, Liking, Dominance)
No. of participants	32
Sampling rate	128 Hz

3.2 Feature Extraction

Fast Fourier transformation was performed for feature extraction reducing it to final dimensions of (58560,70) from (40,40,8064) hence resulting in faster training as well as giving better accuracy.

These extracted features comprise of five frequency bands: Delta-δ (1–4 Hz), Theta-θ (4–8 Hz), Alpha-α (8–14 Hz), Beta-β (14–31 Hz), and Gamma-γ (31–50 Hz), shown in Fig. 2. Extracted 70 features in total and have used PyEEG python library.

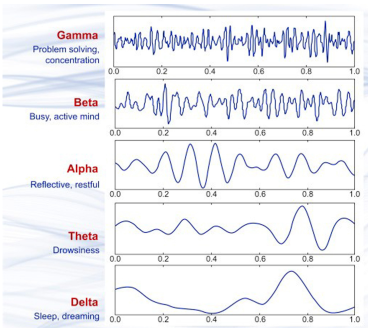

Fig. 2. Brain wave samples for different waveforms [21]

FFT is used to change the Signal domain that is the x-axis from time to frequency. It works on the principle of computing the discrete Fourier transform (DFT) of time Series in an efficient way. It makes the calculations easier by calculating the coefficients of the DFT in an iterative manner, which results in the reduction of computational time as well as computational complexity. It also reduces round-off errors associated with the computations.

As shown in Table 2, 14 channels and 5 bands for our model is selected. The window size was chosen 256 which averages the band power to 2 s. The step size is 16 which means that each 0.125 s update once.

Table 2. FFT parameters description

Channel	1, 2, 3, 4, 6, 11, 13, 17, 19, 20, 21, 25, 29, 31
Bands	4, 8, 12, 16, 25, 45
Window size	256
Step size	16

3.3 Model Architecture

Two deep learning Architectures for our research, Long Short-Term Memory Networks (LSTMs) and Convolutional Neural Networks (CNNs). The dataset used is the python pre-processed version of DEAP dataset. The models were trained for each emotion-arousal, valence, dominance and liking- separately classifying them on a scale of 0 to 9 with varying train-test splits. Both the models were implemented using Keras (Chollet (2015) [22]) and described below:

3.3.1 Long Short-Term Memory (LSTMs)

Long Short-Term Memory Networks (LSTMs) introduced by Hochreiter and Schmidhuber in 1997 [23] are special kind of Recurrent neural networks (RNN). LSTM's were created as a solution to the short-term memory. They have internal mechanisms called gates which can learn which data in a sequence is important to keep or throw away. Hence stores both short- and long-term input units. This is the major reason why we use LSTM in our research.

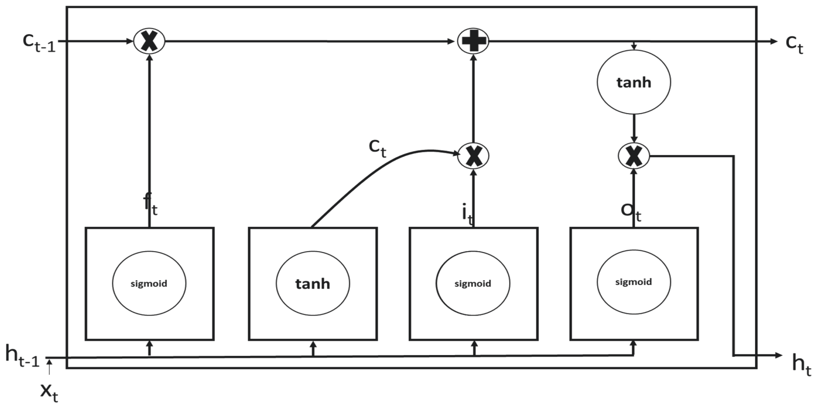

Fig. 3. LSTM cell

A common LSTM unit in Fig. 3 is made of a cell, an input gate, an output gate and a forget gate. The cell remembers values over time and the three gates are used to regulate the flow of information into and out of the cell.

The Sigmoid Activation function helps us to classify if the neuron is active or not. The Sigmoid function transforms a real value to a value ranging from 0 to 1. Consider 0.5 as the threshold value, if the value ranges between 0–0.5 then it is considered not activated, if the value ranges between 0.5–1 then it is considered activated.

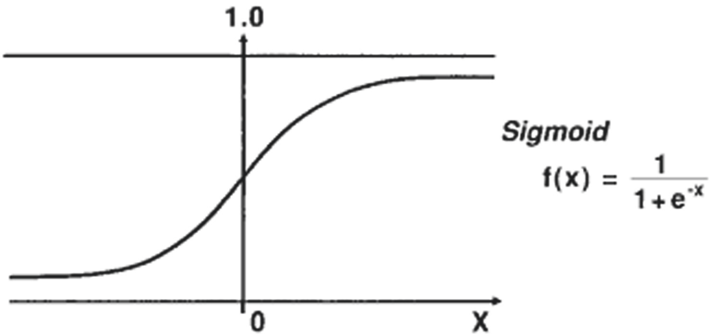

Fig. 4. Sigmoid function [27]

All the three gates use sigmoid function because the gates must give only positive values. The equations for the gates are given:

$$i_t = \sigma\left(\omega_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{1}$$

The first equation is for input gate which tells use what new information will be stored in the cell state.

$$f_t = \sigma\left(\omega_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{2}$$

This second equation is for forget gate; it tells what information to throw away.

$$o_t = \sigma\left(\omega_0\left[h_{t-1}, x\right] + b_0\right) \tag{3}$$

Third equation is for output gate which is used to provide the activation to the final output of LSTM at t timestamp.

i_t: represents input gate
f_t: represents forget gate
o_t: represents output gate
w_x: weight for the respective gate(x)
h_{t-1}: output of previous LSTM block at timestamp t-1
x_t: input at current timestamp
b_x: biases for the respective gate(x)

The next three equations are used for calculation of cell state, candidate cell and the final output.

$$\tilde{c_t} = \tanh\left(\omega_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{5}$$

$$h_t = o_t * tanh(c_t) \tag{6}$$

c_t: cell state (memory) at time stamp(t).
$\tilde{c}_t$: represents candidate for cell state at timestamp(t).
$*$: represents the element wise multiplication of the vectors.

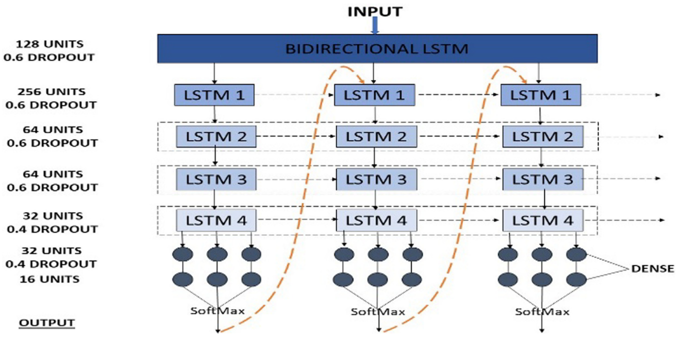

Fig. 5. Proposed LSTM architecture

In our proposed approach one bi-direction LSTM layer, four LSTM layers and two dense layers in the model architecture as shown in the Fig. 4. The first bi-directional LSTM layer has 128 units (in total 256). It involves duplicating the first LSTM layer in the network so that there are two layers side by side. It provides the input sequence as input to the first and a reverse copy of it to the second. Followed by this is the dropout layer with a probability of 0.6. This helps in preventing overfitting by randomly setting inputs to 0 according to the rate during training.

The next layer is a LSTM layer of 256 neurons, followed by dropout layer of 0.6. The next 4 layers are 2 LSTM layers of 64 neurons each followed by a dropout layer. The dropout rates being 0.6 and 0.4 respectively. The final LSTM layer is of 32 neurons followed by dropout layer of 0.4. Then a dense layer of 16 units is used. The activation used for the same is ReLU. Then a dense layer of 10 classes is used with the SoftMax activation function. It results in a multiclass probability distribution over our 10 classes. Knowing the probabilities of all the classes, use of argmax to find the class output is done.

3.3.2 Convolutional Neural Network (CNN)

CNNs are very effective models for Image Processing and classifications. The best thing with the CNN architecture is that there is no need of external feature extraction. The network employs a mathematical operation called convolution. It is a special type of linear equation. Instead of the general matrix multiplication, it uses convolution of data and filters to generate various features which are then passes to the next layer. It is used for its excellent capability to extract spatial features from the data.

$$Z = X * f \tag{7}$$

With X as input, f as filters and $*$ as convolution.

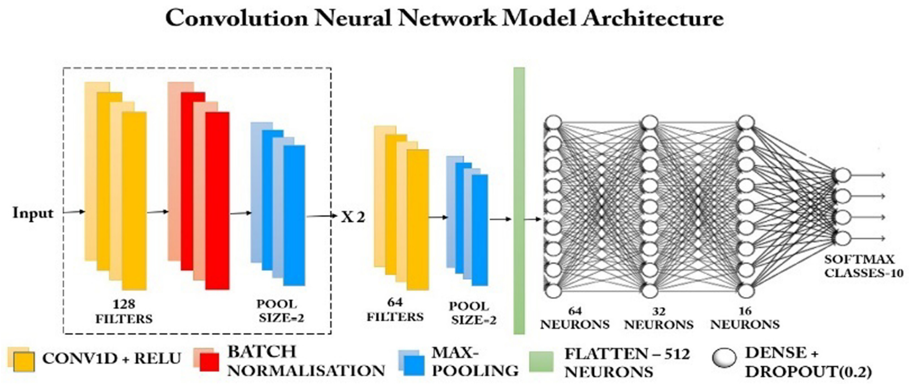

Fig. 6. Proposed CNN architecture

In our proposed model three conv1D, three fully connected dense layers and a dense layer with SoftMax activation for 10 classes in the end as seen in the model architecture in the Fig. 5.

The first convolution layer uses Rectified linear unit (ReLU) as activation function and 128 filters with kernel size of 3. The accurate no. of filters and size of filters is found after a lot of hyperparameter tuning using Grid Search and manual changes. The input passed to the first layer of conv1D is of shape (70,1) and same padding and stride of 1 is used.

The outputs of the first layer are standardized that is having a mean of zero and standard deviation of one using a Batch Normalized layer. The next layer is Max pooling 1D layer of pool size 2 for down sampling the input by taking the maximum value over window size of 2. The padding and strides are set to default i.e. "valid" and "none" respectively. The resulting output has a shape of:

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \tag{8}$$

n_{in}: number of input features
n_{out}: number of output features

k: convolution kernel size
p: convolution padding size
s: convolution stride size

The next Convolution layer is same as the first one followed by batch normalization and max pooling layers. Then Flatten the shape to form a 1-dimensional layer and feed it to a fully connected layer of 64 neurons and TanHyperbolic (tanh) as the activation function. Dropout on the outputs of dense layer is used to reduce the overfitting of the network, with a dropout probability of 0.2. This is followed by 1 dense layer of 32 neurons, tanh activation and dropout layer of 0.2 and another dense layer of 16 neuron with activation function as ReLU and dropout probability of 0.2.

Finally, a Dense layer of 10 neurons with activation function as SoftMax is used to give the output of the network.

3.4 Hyperparameter Tuning

A lot of hyperparameter tuning was carried out while finalizing the Network parameters. Due to the problems like vanishing and exploding gradients Recurrent neural networks (RNNs) is not opted for this. However, one layer of Gated recurrent units (GRU) was tried instead of the Bidirectional LSTM as the first layer, but results were not satisfactory.

In the CNN architecture, Conv1D layers are used because it is most suitable for time series data. Both Max pooling and average pooling was tried but, max pool gave better results as expected from the literature.

Other parameters like the number of epochs, batch size, optimizer, loss function, activation functions and learning rates were finalized using the Grid Search.

Table 3. Network parameters for CNN and LSTM

Parameters	Chosen values
Epoch	200
Batch size	256
Loss function	Categorical cross entropy
Optimizer	Adam
Metrics	Accuracy

The epoch size finalized for the CNN and LSTM architecture is 200 with batch size of 256. The models are trained on various train test splits like 80–20 and 75–25 and K-fold cross validation with 10 folds is also used for finding the most appropriate metrics-accuracy. The loss function used by them for updating the weights during back-propagation is categorical cross entropy and the optimizer used is Adam. And activation function for the last layer is SoftMax for both.

Parameters like no. of layers, number of hidden units, filter size, number of filters and pool size for CNN model and number of hidden neurons, dropout rates and layers for LSTM model were finalized separately, parameters detailed in Table 3. This was done through both Grid search and manual testing.

3.5 Implementation Tools

The environment used for the computation of CNN and LSTM classifiers is Google Collaboratory which is a hosted Jupyter notebook service that requires no setup to use while providing free access to computing resources including GPUs. The python version is Python (3.6.9). The TensorFlow version is 2.2.0. The code is executed in a virtual machine. The GPUs available in CoLab often include Nvidia K80s, T4s, P4s and P100s.

4 Experimental Results

In this section discussion on the experimental results and conclusions attained from the above proposed methodology is done. Created various models with different model architectures and also varying the train test split into different ratios. As Table 4 illustrates that LSTM model architecture which is proposed gives best test accuracy of 88.6% with 75–25 train test split, whereas CNN model architecture gives best accuracy of 87.72 with 80–20 train test split.

Table 4. Comparison between model results on the basis of splits

Classifier	Train-test split	
	75–25	80–20
CNN	87.45%	87.72%
LSTM	88.6%	85.74%

Table 5. Both model's performance on all the four emotions

Classifier	Emotion	Train-test split	
		75–25	80–20
LSTM	Arousal	81.91%	85.07%
	Valence	84.39%	83.83%
	Dominance	69.69%	81.43%
	Liking	88.60%	85.74%
CNN	Arousal	84.77%	85.48%
	Valence	85.01%	82.59%
	Dominance	85.50%	83.61%
	Liking	87.45%	87.72%

The above results are for the liking emotion. Trained these models on all the four emotions individually and got impressive results. Our models were generalizing results very well as they have achieved above 80% accuracy while classifying each emotion. Categorical cross entropy is used as the loss function.

As illustrated in Table 5 both CNN and LSTM model test accuracies are found out for each emotion using both the train test splits and table helps to summarize the results for each emotion. As inferred from the table both the model architectures generalize results very well for all four emotions.

The change in train-test split hardly changes the model performance as the model generalizes results pretty well for both the splits. However, after analyzing the results here this can be concluded that CNN model results are quite précised for each emotion whereas LSTM model results vary with dominance emotion classified with only 69.69% whereas Liking emotion classified with 88.6%.

After training a lot these hyper tuning parameters are finalized to obtain these results. Initially simple LSTM layers are used but the model accuracy was not improving above 65%. Trained model consisting of GRU units but the results were not convincing. Batch size does not have a strong impact on the model results. Dropout and batch normalization layers have significant impact on model's accuracy, dropout helped to avoid overfitting on training data which helped to improve model's results. Categorical cross entropy is used as loss function.

As illustrated in Fig. 6 Similar output are obtained for all the four emotions where least number signifies that emotion present is least and maximum number signifies that the amount of emotion is maximum.

Table 6. Test loss for both models using different train test split

Classifier	Train-test split	
	75–25	80–20
CNN	0.474	0.459
LSTM	0.399	0.503

As illustrated in Table 6. It is found that here LSTM model with data split in 75–25 ratio provided the lowest classification loss of 0.399. Ideal value for categorical cross entropy loss should be equal to zero but practically loss under 1 is considered that the model generalizes results on unseen data pretty well. Both the models reported the value of loss less than 1, which is what we expect as referred to literature. Various learning curves for both the models are also plotted.

All the curves follow the expected pattern as referred to literature, the train and test accuracy for both the models increases with the increase in the number of epochs as shown in Fig. 7. (a) (b). Both the train and test loss decrease with the increase in the number of epochs. Third set of curves is plotted between test accuracy and test loss here. Observed a slight difference between curve of LSTM and CNN as illustrated in Fig. 7. (b), (e). LSTM starts learning a little later than the CNN model and also loss reaches to

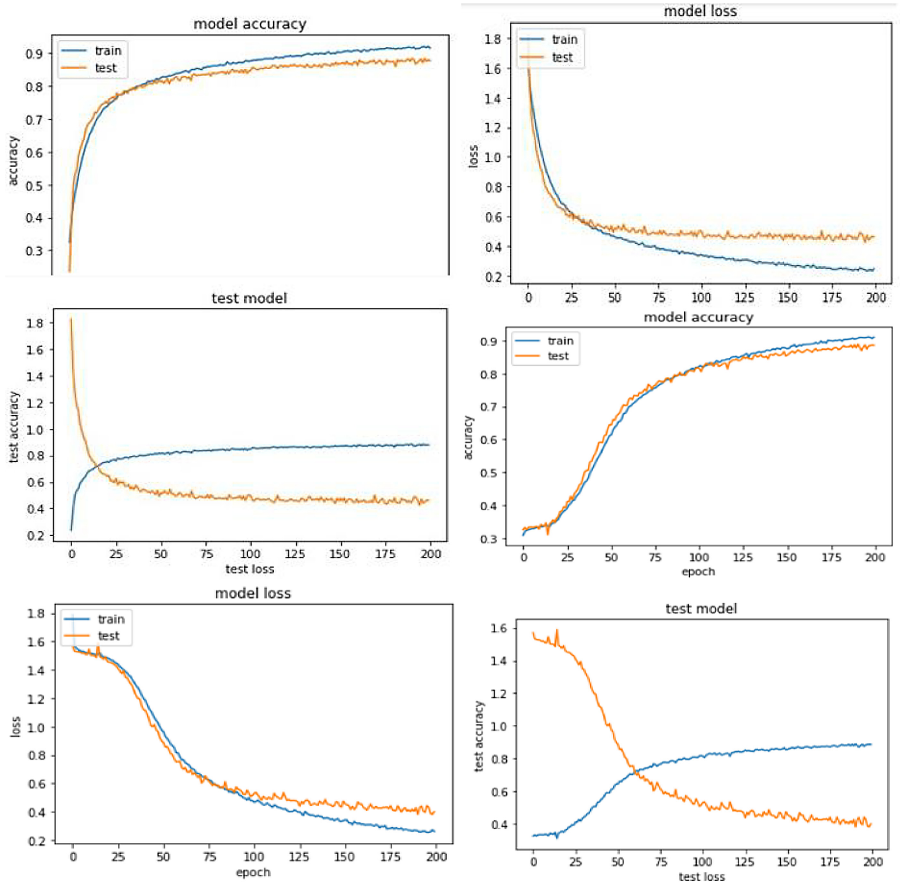

Fig. 7. Row 1: (Left-Right) (a) accuracy vs epoch (CNN), (b) loss vs epoch (CNN) **Row 2: (Left-Right)** (c) test accuracy vs test loss (CNN) (d) accuracy vs epoch (LSTM) **Row 3: (Left-Right)** (e) loss vs epoch (LSTM) (f) test accuracy vs test loss (LSTM)

minimum or stable point after around 40 epochs for CNN whereas it took around 120 epochs to do so for LSTM model.

The comparison between our proposed models and previous works accuracy for different type of emotion is shown in Table 7. The results provided by the proposed method has been compared with four different methods which all used DEAP dataset. The proposed models i.e. our CNN model has attained 87.72% for liking class and also LSTM model has attained 88.6% for the same which is better than the previously attained best accuracy of 87.9% by LSTM model [12] and 81.46% by CNN model [9]. Our proposed models have generalized very well as they are getting above 80% on all classes of emotions. The method proposed by S. Alhagry [12] have managed to get better accuracy than our models for two classes i.e. Arousal and Valence but our proposed models give more finer results i.e. on the scale of 0 to 9 as shown in Fig. 8 for each emotion as compared to S. Alhagry [12] i.e. High or low. This proves that though the accuracy sees a little less, still our model is better capable of classifying the emotions on a finer range.

Table 7. Comparison with other State-of-the-art methods

	Arousal	Valence	Dominance	Liking
Choi et al. [24]	74.65%	78%	-	-
Naser et al. [25]	66.2%	64.3%	-	70.2%
Rozgic et al. [26]	76.9%	68.4%	-	-
S Alhagry et al. [12]	85.65%	85.45%	-	87.9%
Our CNN	84.7%	85.01%	85.5%	87.72%
Our LSTM	81.91%	84.39%	69.7%	88.6%

Figure 8 compares the mean accuracy comparison with other State-of-the-art results. We have compared our results with best results known to us in two (high/low) or three class (high/normal/low) classification of EEG signals on the Arousal, Valence, Dominance, and Liking in Table 7 and Fig. 8.

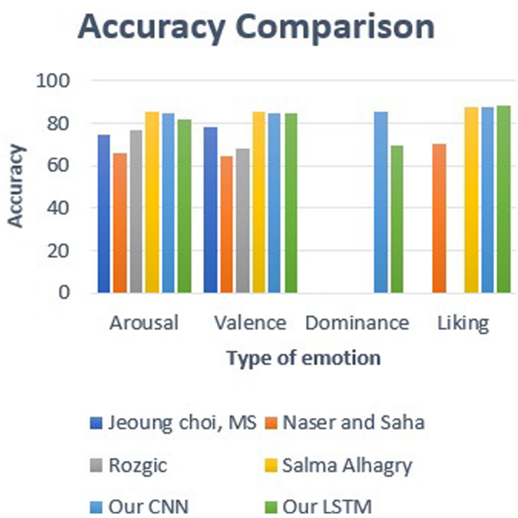

Fig. 8. Mean accuracy comparison with other State-of-the-art results

Confusion matrix is a good technique to measure the performance of classifier for multiclass classification. It is always drawn between true and predicted label. Findings stated that both the classifier algorithms perform similarly for all the classes with classes 4 and 5 being predicted with highest scores. On comparing scores from both Fig. 9 and Fig. 10 the confusion matrix it can be reported that LSTM model scores are better than the CNN model for all the classes except for classes 0 and 7. This also justifies why LSTM architecture is performing better on unseen data (88.6%) than CNN architecture (87.45%).

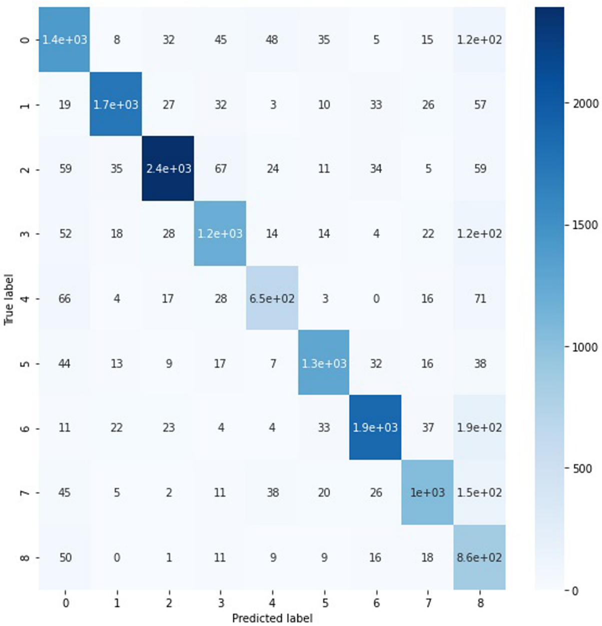

Fig. 9. Confusion matrix for CNN Model

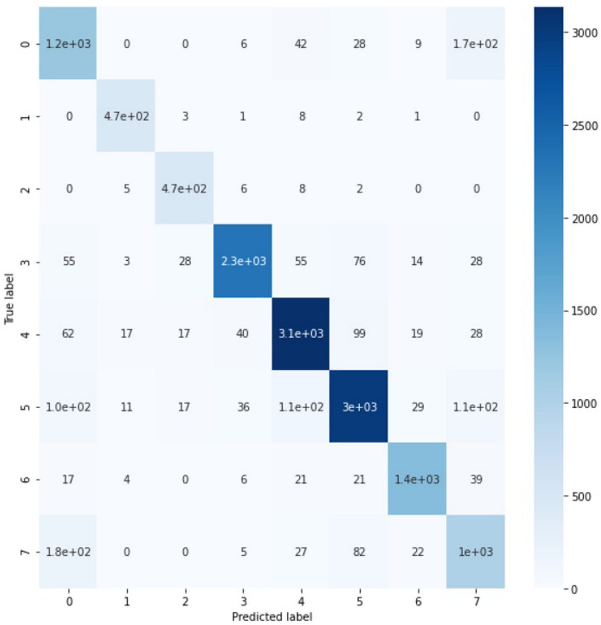

Fig. 10. Confusion matrix for LSTM model

5 Conclusion, Limitation and Future Scope

In this research paper the comparison between CNN and LSTM models on 80–20 and 75–25 Train-Test splits for the classification of EEG signals for emotion analysis is performed successfully. The best result achieved is with the LSTM model using one Bidirectional layer. It has enhanced the performance of the model on this sequence classification problem as compared to the previous papers due to Bi-LSTM's ability to preserve information from future to past as well from past to future. FFT for feature extraction has also contributed in the hike of accuracy as compared to traditional feature extraction techniques. CNN also gave a good accuracy due to its capability to extract spatial features from the input signals.

Still some of the limitations lead to setback in integrating real-life emotion analysis systems like collecting emotion data in a convenient and reliable way such that the signal to noise ratio is increased. Also sometimes decoding the affective state consistently and accurately pose problems. Real-Time online analysis systems also require timely output of the EEG data computation, imposing constraints on the computation speed. The collection of EEG data with the help of EEG headsets with either dry or wet electrodes pose real problems including discomfort to the users especially with long hair, its weight on the head, etc. Sometimes decoding the affective state consistently and accurately pose problems.

In future our team would like to work on Multi-task cascaded-hybrid models of LSTM and CNN to combine their power and thus increase the accuracy of the emotion recognition system. The emotion detection system can be very effective in eliminating the gap between human emotions and computational technology, thus, enabling robots and computers to receive natural emotional feedback and improve human experiences. Even this system can be used by therapist to better evaluate their patients and find ways to predict and prevent depression before there are any clear outward signs of it.

Acknowledgment. This research work is performed under the nation wise initiative leadingindia.ai and Bennett University, India. They have supported us with lab and equipment during the experiments.

References

1. Acharya, D., et al.: An enhanced fitness function to recognize unbalanced human emotions data. Expert Syst. Appl. **166**, 114011 (2020)
2. Acharya, D., Billimoria, A., Srivastava, N., Goel, S., Bhardwaj, A.: Emotion recognition using fourier transform and genetic programming. Appl. Acoust. **164**, 107260 (2020)
3. Bairavi, K., Sundhara, K.K.: EEG based emotion recognition system for special children. In: Proceedings of the 2018 International Conference on Communication Engineering and Technology, pp. 1–4, February 2018
4. Dabas, H., Sethi, C., Dua, C., Dalawat, M., Sethia, D.: Emotion classification using EEG signals. In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 380–384, December 2018
5. Li, Y., Hu, B., Zheng, X., Li, X.: EEG-based mild depressive detection using differential evolution. IEEE Access **7**, 7814–7822 (2018)

6. Li, X., Yan, J.Z., Chen, J.H.: Channel division based multiple classifiers fusion for emotion recognition using EEG signals. In: ITM Web of Conferences, vol. 11, p. 07006. EDP Sciences (2017)
7. Donmez, H., Ozkurt, N.: Emotion classification from EEG signals in convolutional neural networks. In: 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1–6. IEEE (2019)
8. Yang, Y., Wu, Q., Qiu, M., Wang, Y., Chen, X.: Emotion recognition from multi-channel EEG through parallel convolutional recurrent neural network. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE, July 2018
9. Tripathi, S., Acharya, S., Sharma, R.D., Mittal, S., Bhattacharya, S.: Using deep and convolutional neural networks for accurate emotion classification on DEAP dataset. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 4746–4752, February 2017
10. Dunne, R.A., Campbell, N.A.: On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In: Proceedings of 8th Australian Conference on the Neural Networks, Melbourne, vol. 181, p. 185. Citeseer, June 1997
11. Liu, W., Zheng, W.-L., Bao-Liang, L.: Emotion recognition using multimodal deep learning. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) Neural information processing, pp. 521–529. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46672-9_58
12. Alhagry, S., Fahmy, A.A., El-Khoribi, R.A.: Emotion recognition based on EEG using LSTM recurrent neural network. Emotion **8**(10), 355–358 (2017)
13. Zhang, J., Chen, M., Hu, S., Cao, Y., Kozma, R.: PNN for EEG-based emotion recognition. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 002319–002323. IEEE, October 2016
14. Zhong, P., Wang, D., Miao, C.: EEG-based emotion recognition using regularized graph neural networks. IEEE Trans. Affect. Comput. (2020)
15. Li, Y., et al.: A novel bi-hemispheric discrepancy model for EEG emotion recognition. IEEE Trans. Cogn. Dev. Syst. (2020)
16. Acharya, D., Goel, S., Bhardwaj, H., Sakalle, A., Bhardwaj, A.: A long short term memory deep learning network for the classification of negative emotions using EEG signals. In: 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom, pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9207280
17. Bhardwaj, A., Tiwari, A., Varma, M.V., Krishna, M.R.: Classification of EEG signals using a novel genetic programming approach. In: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO Comp 2014), pp. 1297–1304. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2598394.2609851
18. Bhardwaj, H., Sakalle, A., Bhardwaj, A., Tiwari, A.: Classification of electroencephalogram signal for the detection of epilepsy using Innovative Genetic Programming. Expert Syst. **36**, e12338 (2019). https://doi.org/10.1111/exsy.12338
19. Acharya, D., Goel, S., Asthana, R., Bhardwaj, A.: A Novel fitness function in genetic programming to handle unbalanced emotion recognition data. Pattern Recogn. Lett. **133**, 272–279 (2020). https://doi.org/10.1016/j.patrec.2020.03.005
20. https://www.eecs.qmul.ac.uk/mmv/datasets/deap/readme.html
21. Abhang, P.A., Mehrotra, S.C.: Introduction to EEG- and Speech-Based Emotion Recognition. Chapter 2 - Technological Basics of EEG Recording and Operation of Apparatus (2016)
22. Chollet, F. (2017). Keras (2015)
23. Hochreiter, S., Schmidhuber, J.: LSTM can solve hard long time lag problems. In: Advances in Neural Information Processing Systems, pp. 473–479 (1997)

24. Choi, E.J., Kim, D.K.: Arousal and valence classification model based on long short-term memory and deap data for mental healthcare management. Healthc. Inf. Res. **24**(4), 309–316 (2018)
25. Naser, D.S., Saha, G.: Recognition of emotions induced by music videos using DT-CWPT. In: 2013 Indian Conference on Medical Informatics and Telemedicine (ICMIT), Kharagpur, pp. 53–57 (2013). https://doi.org/10.1109/IndianCMIT.2013.6529408
26. Rozgić, V., Vitaladevuni, S.N., Prasad, R.: Robust EEG emotion classification using segment level decision fusion. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, pp. 1286–1290 (2013). https://doi.org/10.1109/ICASSP.2013.6637858
27. Næs, T., Kvaal, K., Isaksson, T., Miller, C.: Artificial neural networks in multivariate calibration. J. Near Infrared Spectrosc. **1**(1), 1–1 (1993). https://doi.org/10.1255/jnirs.1