NORTH SOUTH UNIVERSITY

Emotion Analysis using Deap Learning and Machine Learning Techniques
CSE499A
Supervisor: Dr. Shafin Rahman

A DISSERTION
SUBMITTED TO THE DEPARTMENT OF
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
OF NORTH SOUTH UNIVERSITY
IN THE PARTIAL FULFILMENT OT THE REQUIREMENTS
FOR THE DEGREE OF NORTH SOUTH UNIVERSITY IN

Date
19$^{\text{th}}$ April 2020, Sunday

Declaration

It is hereby acknowledged that:

- No illegitimate procedure has been practiced during the preparation of this document.
- This document does not contain any previously published material without proper citation.
- This document represents our own accomplishment while being Undergraduate Students in the **North South University**

Sincerely,

Student 1: Erfan Mostafiz
1912734042, Signature

Student 2: Shukdev Datta
1911838042, Signature

Student 3: Md Toufique Husein
1921750642, Signature

Approval

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 1 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 2 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 3 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 4 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 5 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 6 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 7 Name
Signature

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

Officer 8 Name
Signature

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) and also machine learning techniques like KNR, Bagging Regressor, Random Forest Regressor, SVR, AdaBoost. We have used the state of art DEAP dataset which contains EEG emotion data of 32 participants for 40 trials each. The best result was obtained from CNN model with an accuraacy of 87.6% on the Liking Class of emotion. LSTM also gave a good accuracy of 87%. The best result for deep learning model was obtained from LSTM classifier with an accuracy of 88.6the liking emotion. CNN also gave a good accuracy of 87.72to extract spatial feature from the input signals. Thus, both these models are quite beneficial in this context. On the other hand, for the machine learning models, K-nearest neigbors regressors have a whopping 93.9% accuracy on the Liking class with accuracies on the other 3 classes also greater than 90%.

Contents

List of Figures

List of Tables

1 Introduction

Emotions are mental states brought on by neuro physiological changes, variously associated with thoughts, feelings, behavioral responses, and a degree of pleasure or displeasure. In recent years, emotion analysis has been a great research topic and there have been many studies related to Affective Computing by the use of Deep Learning. Nowadays there vast is concern of emotion analysis in medical, education, gaming, product, marketing fields. In the era of pandemic like Covid-19, people have been undergoing emotion changes from time to time. So, it has been a great concern for health sector to analyze emotions of the patients and common people via emotion analysis. Now how do we identify the emotion state of a person? We have used EEG data to identify and predict emotions. According to studies, physiological signals tend to reflect people's real emotions much more accurately than facial expressions, postures or voice. Physiological signals such as Electrooculography (EOG), Electrocardiogram (ECG) and Electromyography (EMG) are indirect reactions caused by emotions, although they are far better than facial expressions, postures or voice, since the generation of emotion is closely related to the cerebral cortex of the brain. Convolution Neural Network (CNN) and Long Short Term Memory (LSTM) of Deep learning have been used in our work. Also several machine learning regressors were also implemented - KNR, Bagging Regressor, Random Forest Regressor, SVR, AdaBoost.

2 Related Works

Many researchers worked on Emotion analysis, for example, Deep neural networks (DNN) and convolutional neural networks (CNN) for emotion categorization on the DEAP dataset were investigated by S Tripathy et al. in 2017 [1]. Their DNN model's suggested architecture consists of a 4040-unit input layer, followed by three hidden layers with 5000, 500, and 1000 units each. Depending on the need, the output layer uses a 2 or 3-class softmax classifier [2]. Two convolutional layers, Maxpooling, and Dropout layers, which connect to Fully Connected layers to generate the output, were also included in the proposed CNN architecture.In terms of valence and arousal classification using two classes (high and low), they reach the accuracy of 75.78% and 73.125% for DNN and 81.406% and 73.36% for CNN, respectively. The accuracy achieved for valence and arousal's three categorization classes (high, normal, and low) is 58.44% for DNN and 55.70% for CNN, respectively

A Regularized Graph Neural Network (RGNN) and two regularizers, Node-wise Domain Adversarial Training (NodeDAT) and Emotion-Aware Distribution Learning are proposed by P. Zhong et al. (2020) [3] for EEG-based emotion identification (EmotionDL). With average accuracies of 94.24% and 79.37% for subject-dependent classification accuracy on SEED and SEED-IV (all bands), they outperformed the state-of-the-art results of the bi-hemispheric discrepancy model (BiHDM) proposed by Y.Li et al. (2019) [4], which had average accuracies of 93.12% and 74.35%. Additionally, RGNN achieved mean accuracy in subject-independent classification of 85.30% and 73.84%, respectively.

According to W. Liu et al. [5], the Bimodal Deep Auto-encoder extracts features (BDAE). They create two Restricted Boltzmann Machines (RBM), one for eye movement characteristics and the other for EEG (EEG RBM) (eye RBM). They combine the concealed layers to produce an upper RBM. The high-level characteristics extracted are utilized to train a linear SVM classifier after being selected using the BDAE network. On the SEED and DEAP datasets, the mean accuracies attained with the BDAE network are 91.01% and 83.25%, respectively.

In addition to providing an LSTM architecture for the assessment of negative emotions, D. Acharya et al. [6] also present a brief overview of human behavior across various age and gender groups. With data split 50-50, 60-40, 70-30, and 10-fold cross-validation, their LSTM model for four classes of negative emotion classification achieves classification accuracy of 81.63%, 84.64%, 89.73%, and 92.84%. Both the DEAP and SEED datasets have been used to evaluate the models.

Schmidt et al. employed music to elicit four different emotions [7]. They discovered that while utilizing positive musical materials, the frontal areas of the left hemisphere experienced heightened EEG activity, and when using negative musical materials, the frontal portions of the right hemisphere experienced enhanced EEG activity. The authors come to the conclusion that the frontal lobes of the human brain and emotion are closely related. They obtained A 0.67.7% classification accuracy.

Research on emotion identification based on physiological cues was conducted by Lu and his colleagues [8]. They have recently established an accessible EEG-based emotion identification database. The binary (happy vs. calm) classification rate for the emotion-related database created by additional researchers, which consists of four physiological signals from the ECG, galvanic skin response, skin temperature, and respiration, achieved 86.7% Accuracy [9].

A reported categorization accuracy of roughly 55%. In order to provide stimulation, Koelstra et al. gave each of the 32 subjects instructions to watch 40 different pieces of music video content. They then recorded the self-report (subjective ratings), facial expression, EEG, and peripheral physiological signals [10].

Based on the DEAP dataset, S. Alhagry et al. (2017) [11] presented a 2-layer stacked LSTM architecture. 64 units with the ReLU activation function make up the initial LSTM layer, followed by a dropout layer with a 0.2 probability. A dense layer with sigmoid activity is connected to the second layer, which consists of 32 neurons with this function. They divide valence, arousal, and liking into high and low classes and, using the DEAP dataset, correspondingly get average accuracy of 85.65%, 85.45%, and 87.99%.

3 Methodology

The research approach used in this work is detailed in Figure 3.1

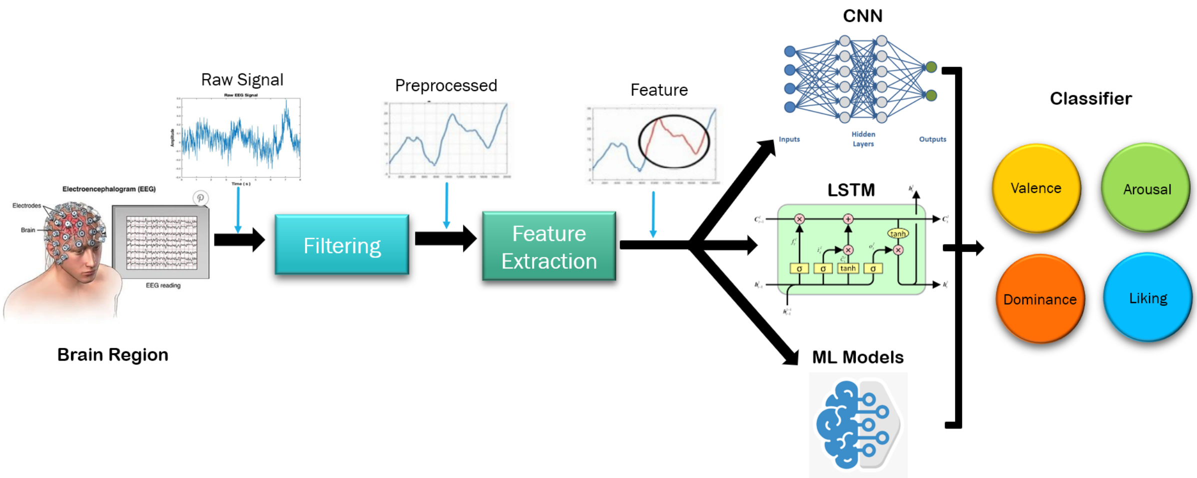

Figure 3.1: Flow Chart of our Research Methodology

3.1 Dataset Details

In our research work, we have used a state-of-the-art dataset - DEAP Dataset, which is a multimodal dataset for the analysis of human affective states. The dataset details is given in Table 3.1.The EEG and peripheral physiological signals of 32 participants were recorded as each watched 40 1-minute-long excerpts of music videos. Participants rated each video in terms of the levels of arousal, valence, like/dislike, dominance, and familiarity.

Dataset link: https://www.eecs.qmul.ac.uk/mmv/datasets/deap/

Each of these 32 preprocessed.dat files collected from the DEAP dataset page contained two arrays: data and labels, one for each of the 32 participants. Data has dimensions of 40,40,8064. Each video had 40 channels and had 8064 EEG signal data points, for a total of 322560. The labels were in the shape of a 40 by 4, where 4 stands for valence, arousal, dominance, and liking. NumPy arrays are loaded and used in Python.

data files encoded in Latin-1 using the pickle library.

Criterion	**DEAP Dataset**
Dataset Size	22 subjects x 40 trials
Stimulus duration	60s
Modality	Vision, EEG, Peripheral
Used bio signals	EEG, EOG, EMG, GSR, Respiration belt, Plethysmograph, Temperature
Labels	Continuous valence and arousal from 1 to 9

Table 3.1: Details of the DEAP Dataset

3.2 Feature Extraction

Fast Fourier transformation (FFT) is used as a feature extraction technique for our proposed work. It reduces the dataset dimension from (40,40,8064) to (58560,70) dimension. Because of this our models CNN and LSTM provided better accuracy with faster execution.

FFT (Fast Fourier transformation) has been used to transform the signal from Time Domain to Frequency Domain. It works on the principle of computing the discrete Fourier transform (DFT) of time Series. It makes the calculations smooth using iterative manner to calculate the coefficients of the DFT, which results in the reduction of computational time as well as computational complexity. It reduces the error related to the computations.

FFT is used to compute DFT (Discrete Fourier Transform) by decomposing a sequence into different frequencies

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \tag{3.1}$$

where $e^{-i2p/N}$ is a primitive Nth root of 1 and there are N outputs X_k, and each output requires a sum of N terms.

These extracted features comprise of five frequency bands: Delta- (1–4 Hz), Theta- (4–8 Hz), Alpha- (8–14 Hz), Beta- (14–31 Hz), and Gamma- (31–50 Hz), shown in Figure 3.2

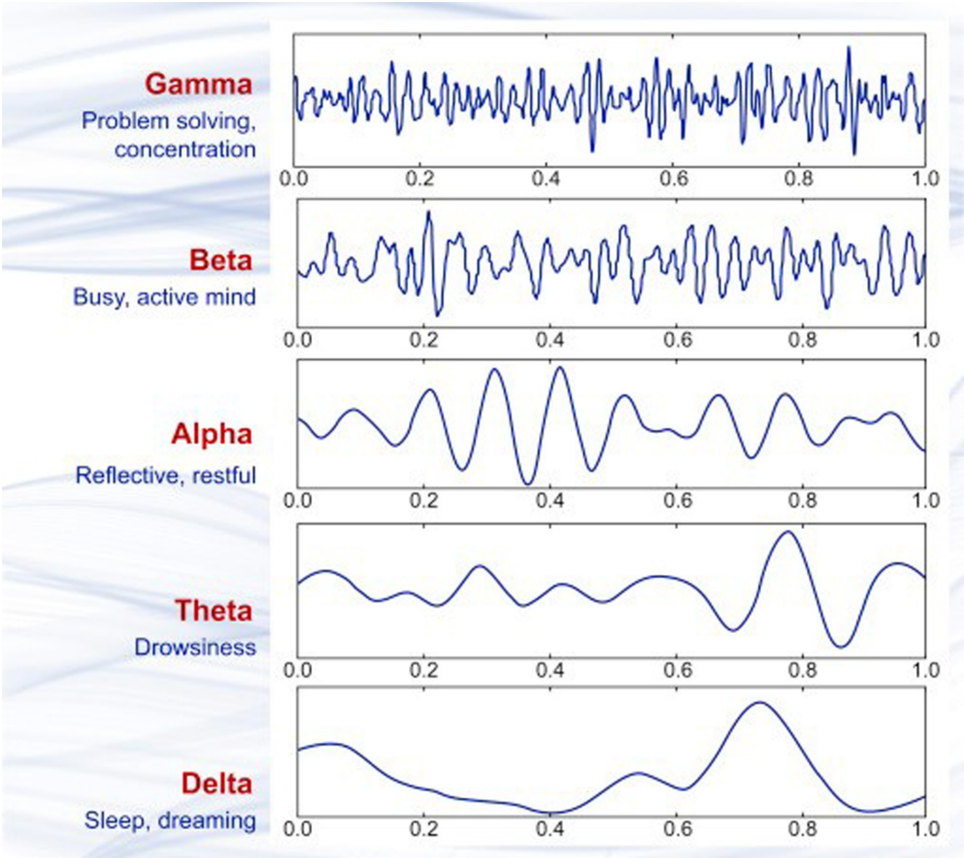

Figure 3.2: Brain wave samples for different waveforms

Table 3.2 shows that we have used 14 channels and 5 bands for our models of LSTM, CNN and Machine learning algorithms. 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.

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

Table 3.2: FFT Parameters used

3.3 Model Architecture

We have used two deap learning models in our work - Long Short Term Memory Network (LSTM) and Convolutional Neural Network (CNN). The models were trained for each emotion, arousal,valence, dominance and liking- separately classifying them on a scale of 0 to 9. Keras and pyTorch libraries have been used to implement the model in Google Colab.

3.3.1 Long Short Term Memory

Long Short Term Memory Network (LSTM), introduced in 1997 [12], is an advanced kind of Recurrent Neural Network (RNN), a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. They have internal mechanisms called gates which can learn which data in a sequence is important to keep or throw away. Hence it stores both short- term and long-term input units.

LSTM consists of 3 parts and each part performs a specific function. The first part chooses whether the information coming from the previous timestamp is relevant and to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp. These 3 parts are known as gates. The first part is called Forget gate, the second part is known as the Input gate and the last part is the Output gate. Figure 3.3 describes these 3 gates of an LSTM cell.

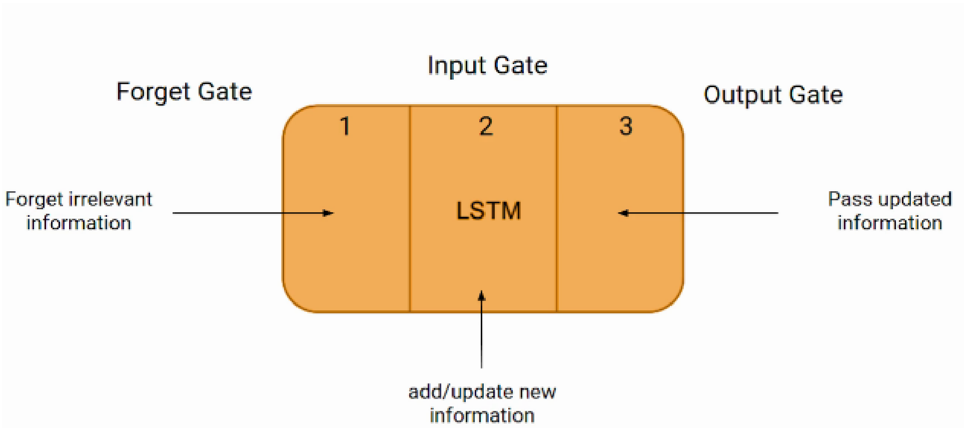

Figure 3.3: The 3 Gates of an LSTM cell

LSTM also has a hidden state, just like RNNs, where H_{t-1} represents the hidden state of the previous timestamp and H_t is the hidden state of the current timestamp. In addition to that LSTM also have a cell state represented by C_{t-1} and $C(t)$ for previous and current timestamp respectively. The hidden state is known as Short Term Memory and the cell state is known as Long Term Memory. These are shown in Figure 3.4.

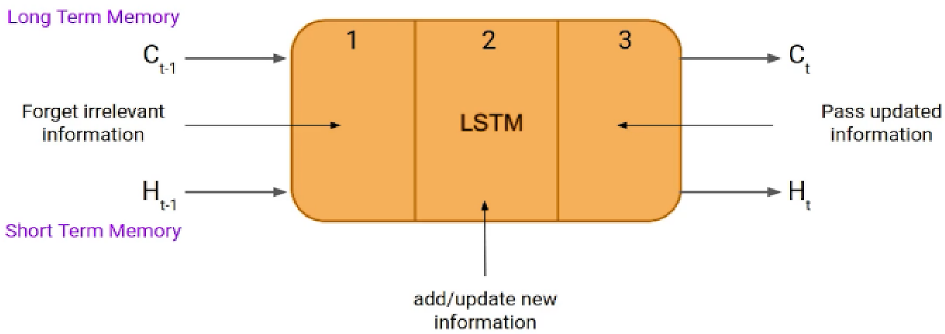

Figure 3.4: Long Term and Short Term Memory of LSTM cell

All the three gates (forget, input and output) use Sigmoid function because the gates must give only positive values. The Sigmoid Activation function [13] 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. Considering 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.

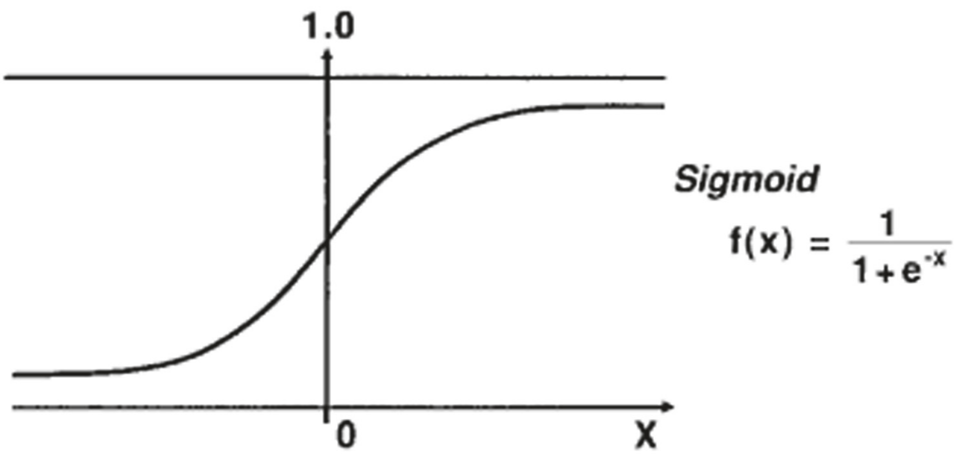

Figure 3.5: Sigmoid Activation Function

The equation of the Input Gate, which tells what new information will be stored in the cell state, is given in Equation 3.2.

$$i_t = \sigma(w_i[h_(t-1), x_t] + b_i) \tag{3.2}$$

The equation for forget gate, which tells what information to throw away, is given in Equation 3.3

$$f_t = \sigma(w_f[h_(t-1), x_t] + b_f) \tag{3.3}$$

The equation for output gate, which is used to provide the activation to the final output of LSTM at t timestamp, is given in Equation 3.4.

$$o_t = \sigma(w_0[h_(t-1), x] + b_0) \tag{3.4}$$

where, 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, and 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.

$$c_t = tanh(w_c[h_(t-1), x_t] + b_c) \tag{3.5}$$

$$c_t = f_t * c_{t-1} + i_t * c_t \tag{3.6}$$

$$h_t = o_t * tanh(c_t) \tag{3.7}$$

where, c_t: cell state (memory) at time stamp(t), $*$: represents the element wise multiplication of the vectors.

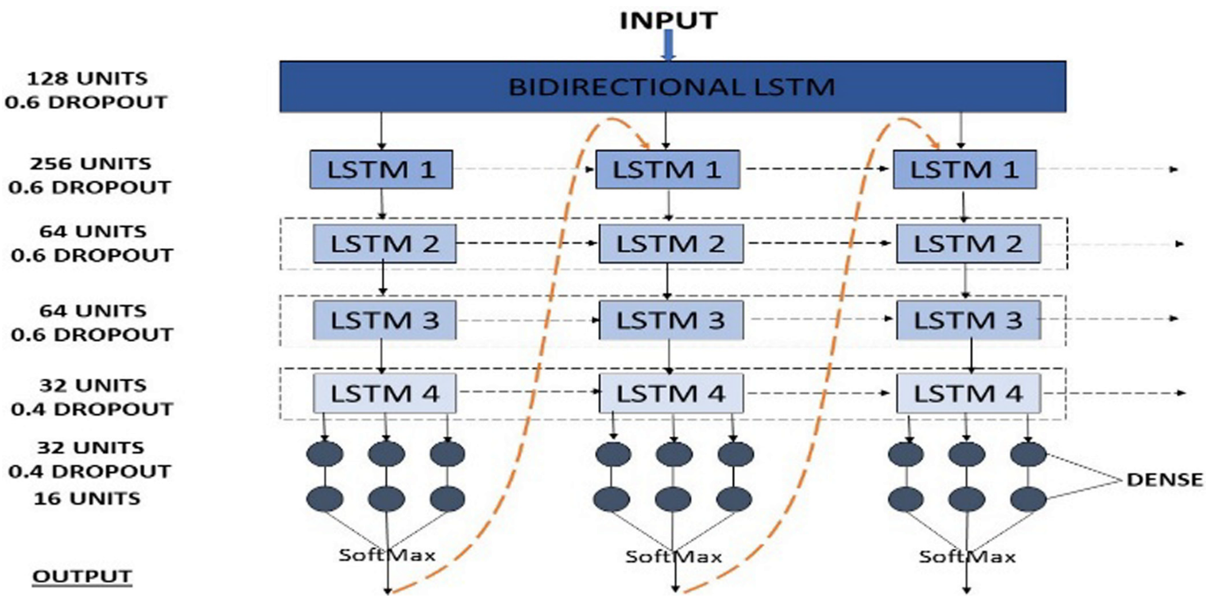

Figure 3.6: Proposed LSTM model architecture

In our proposed LSTM model architecture, described in Figure 3.6, 1 bi-directional LSTM layer, 4 LSTM layers and 2 dense layers have been used. 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 sequence. 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, for the next 2 LSTM layers. The final LSTM layer is of 32 neurons followed by dropout layer of 0.4. Then a dense layer of 16 units is used with ReLU as the activation function.

$$ReLU = f(x) = max(0, x) \tag{3.8}$$

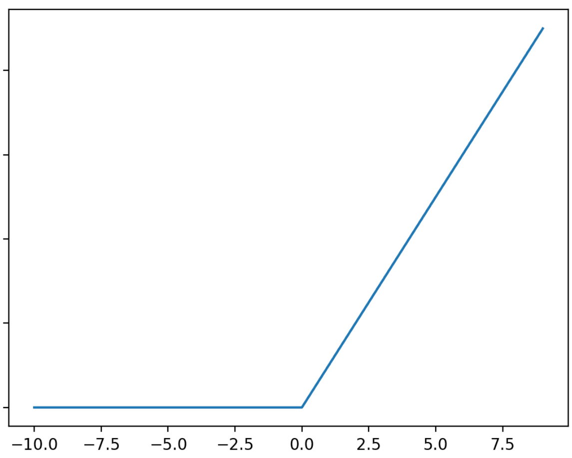

Figure 3.7: ReLU Activation Function

Then a dense layer of 10 classes is used because our final output should be multiclass probability distribution over our 10 classes of emotion taken from the DEAP dataset. SoftMax is used as the activation function for the final output layer.

$$SoftMax = s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{3.9}$$

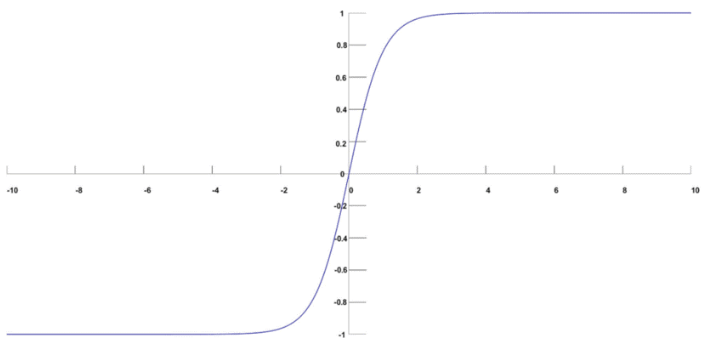

Figure 3.8: SoftMax Activation Function

3.3.2 Convolution Neural Network

Convolution Neural Network (CNN) is one of the most important algorithm in Deep learning that is used for image processing and many other interesting things. The technique that this model uses is known as Convolution and Convolution is a mathematical operation of two functions which will produce a third function that depicts how the shape of one is modified by the other.

$$Z = X * f \tag{3.10}$$

In Equation 3.10, $X = input$, $f = filters$ and $* = convolution$

In our proposed model, we have chosen 10 as the number of classes. SoftMax is used to assign decimal probabilities to each class in a multi-class problem. Also it can be seen that there are three dense layers connected with each other. Moreover, the first CONV1D layer is using ReLU as the activation function along with 128 filters and kernel size 3. ReLU is a piecewise linear function that will show the output of an input if it is positive otherwise it will show zero value for negative inputs. In order to find the correct number of filters and size of the kernel, we used Hyper-parameter optimization like Grid Search to figure out the correct numbers to use. As you can see there is an INPUT in front of the first CONV1D layer which is of shape (70, 1) and padding and strides are both set to 1.

Later the output of the first CONV1D layer is passed on to the Batch Normalized layer where the mean of the output was normalized to zero and standard deviation of the output was normalized to one. The next layer after Batch Normalized layer is known as Max pooling 1D layer that has a pool size of 2 in order to down sample the input by taking the max value over window size of 2 and padding are set to valid and strides are set to none.

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \tag{3.11}$$

In Equation 3.11 n_{in}: number of input features, n_{out}: number of output features, k: convolution kernel size, p: convolution padding size, s: convolution stride size

After Max pooling 1D layer, again the same first CONV1D layers is connected and followed by Batch Normalized layer and Max pooling 1D layer. We used Flatten which will convert the data to a 1D layer and then feed it to the next layer of 64 neurons and tanh as the activation function. TanHyperbolic (tanh) is a sigmoid function that ranges from -1 to +1, The Dropout on the output of the dense layer is used to avoid the overfitting of the network. The 64 neurons dense layer is followed by 1 dense layer of 32 neurons which has tanh [5] as activation function and dropout probability of 0.2 which is again followed by another dense layer of 16 neurons with ReLU as activation function and having dropout probability of 0.2. Finally we used another dense layer of 10 neurons with activation function as Softmax that is used to give the output of the network.

The Diagram in Figure 3.9 shows the architecture of our CNN model.

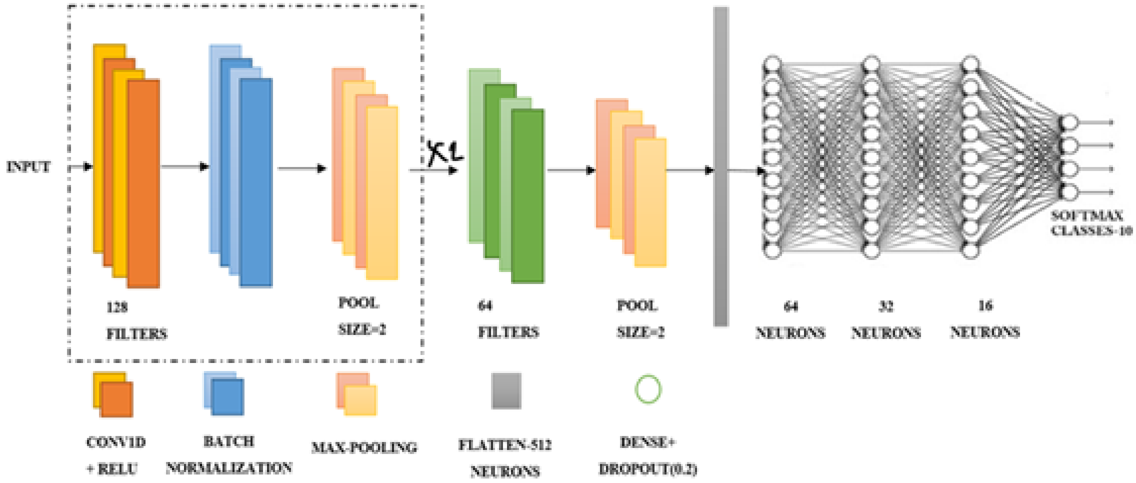

Figure 3.9: Proposed CNN model architecture

3.4 Hyperparameter Tuning

In our CNN model, CONV1D layer was used because it has an advantage of learning from the raw time series data directly. And since we are dealing with time series data, hence CONV1D layer is the most suitable to use. We also tried to use average pooling but it did not give any satisfactory result, so we used Max pooling which gave a satisfactory result. Other parameters that are used for CNN and LSTM are shown in Table 3.3:

Parameters	Chosen values
Epoch	200
Batch size	250
Loss function	Categorical cross entropy
Optimizer	Adam
Metric	Accuracy

Table 3.3: Best Hyperparameters that were tuned

The epoch is set to 200 and Batch size is set to 250 for both LSTM and CNN model. The loss function is used measure how well the neural network has modelled the training data. Optimizer Adam is used and metrics is set to accuracy. All the parameters like hidden layers number, filter numbers etc. are selected through hyper parameter optimization.

3.5 Machine Learning Models

Several Machine Learning models were performed on our dataset to test their evalutaions.

3.5.1 AdaBoost Regressor

AdaBoost Regressor [14] is used for regression problem. This regressor is a meta-estimator (an estimator that takes another estimator as the parameter) which fits the regressor on the original dataset and later fits extra copies of the regressor on the same dataset. The weights of the instances are to be adjusted based on the error of the current prediction and in this way, following regressors can focus more on difficult cases in hand.

The AdaBoost Regressor have parameters like n_estimators=5000 and learning_rate=0.01.

3.5.2 Support Vector Regressor

Support Vector Regressor or SVR in short is used to solve regression problems. SVR allows us to be flexible in case of how much error is acceptable for our model and it fits the data using a hyper-plane.

3.5.3 Random Forest Regressor

A random forest [15] is a meta-estimator that fits a number of decision tree classifiers on several sub-samples of the dataset and uses mean to improve the accuracy of the prediction and control over-fitting. The sub-sample size is controlled with the max_samples parameter, if bootstrap=True (default: bootstrap samples are used), otherwise the whole dataset is used to build each tree.

The Random Forest Regressor have parameters like n_estimators=512 (512 base estimators) and n_jobs=-1 (use all processors).

3.5.4 Bagging Regressor

A bagging regressor is a type of ensemble meta-estimator that fits each base regressors on random subset samples of the original dataset and then take the final decision by aggregating the predictions of each regressor through majority vote or by averaging. The Bagging Regressor have parameters like n_estimators=512 (512 trees in the forest) and n_jobs=-1 (use all processors).

3.5.5 K-Nearest Neighbors Regressor

The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set. The K Neighbors Regressor have parameters like n_estimators=1 (Number of neighbors).

4 Results

This section discusses the experimental findings and findings drawn from the deep learning and machine learning technique mentioned above. Multiple models were built using various model architectures, and the train test split was also changed to reflect various ratios.

4.1 Deep Learning Models' Results

The suggested LSTM model architecture yields the best test accuracy of 87.6%, whereas the CNN model architecture yields the highest accuracy of 87% as shown in Table 4.1. We have trained both our LSTM and CNN models 4 times for the 4 different target classes of emotions - arousal, valence, dominance and liking.

Models	**Arousal%**	**Valence%**	**Dominance%**	**Liking%**
Choi et al.	74.65	78	-	-
Naser et al.	66.2	64.3	-	70.2
Rozgic et al	76.9	68.4	-	-
S Alhagry et al	85.65	85.45	-	87.9
Our CNN	86.2	85.1	80.2	87.6
Our LSTM	86.4	86.2	81.9	87

Table 4.1: Comparison of our Deep Learning models with other state of the art results

Choi et al. and Rozgic et al. both actually performed their model execution on two labels Arousal (74.65% accuracy for Choi et al. and 76.9% accuracy for Rozgic et al.) and Valence (78% accuracy for Choi et al. and 68.4% accuracy for Rozgic et al.) but as we can see they have not performed the model on other two labels Dominance and Liking.

Choi et al. have used LSTM as the model to train the EEG data of the DEAP Dataset. Rozgic et al. used Nearest Neighbor to train the EEG data of the DEAP Dataset.

Naser et al. and S Alhagry et al. both actually performed their model execution on three labels Arousal (66.2% accuracy for Naser et al. and 85.65% accuracy for S Alhagry et al.), Valence (64.3% accuracy for Naser et al. and 85.45% accuracy for Rozgic et al.) and Liking (70.2% accuracy for Naser et al. and 87.6% accuracy for Rozgic et al.) but as we can see they have not performed the model on another label "Dominance".

Naser et al. have used DT-CWPT model to train the EEG data of the DEAP Dataset. S Alhagry et al. have used LSTM-RNN model to train the EEG data of the DEAP Dataset.

Our CNN model have scored more than 80% on all the 4 labels. Also our LSTM model have also scored more than 80% on 4 labels outperforming every other models.

4.1.1 Result Graphs

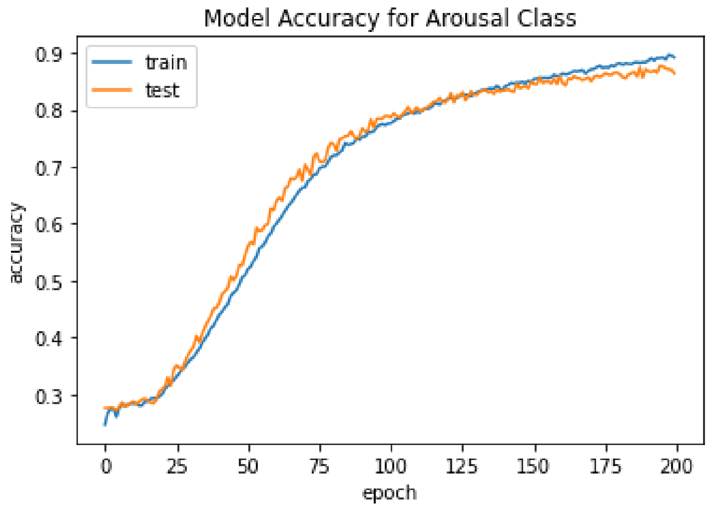

Figure 4.1: LSTM accuracy of arousal class

4.2 Machine Learning Models' Results

Table 4.2 shows the performance of the 5 different machine learning models that we used in predicting all 4 classes of emotions

Among all the regressors we have used, the best one to perform was the KNeigh-

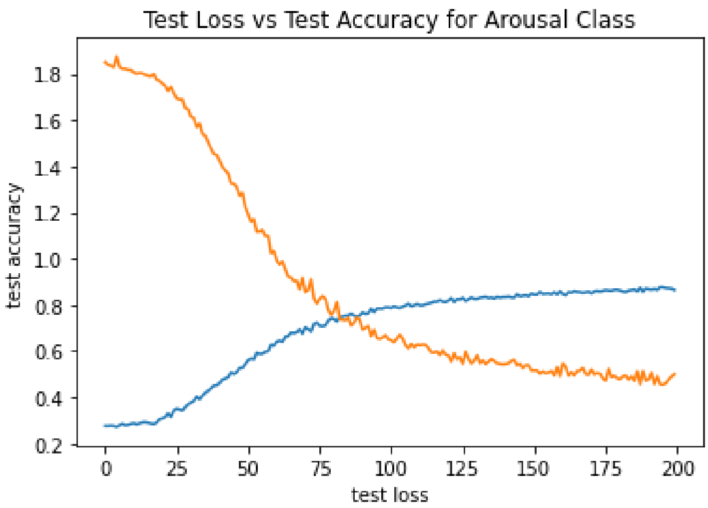

Figure 4.2: LSTM accuracy vs loss of arousal class

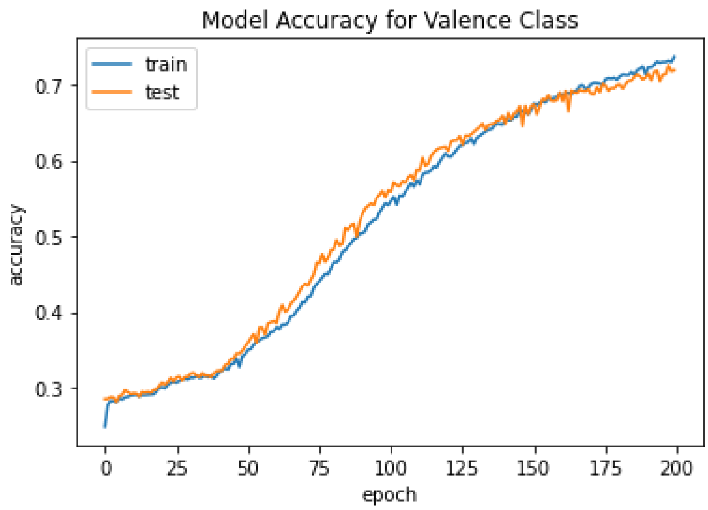

Figure 4.3: LSTM accuracy of valence class

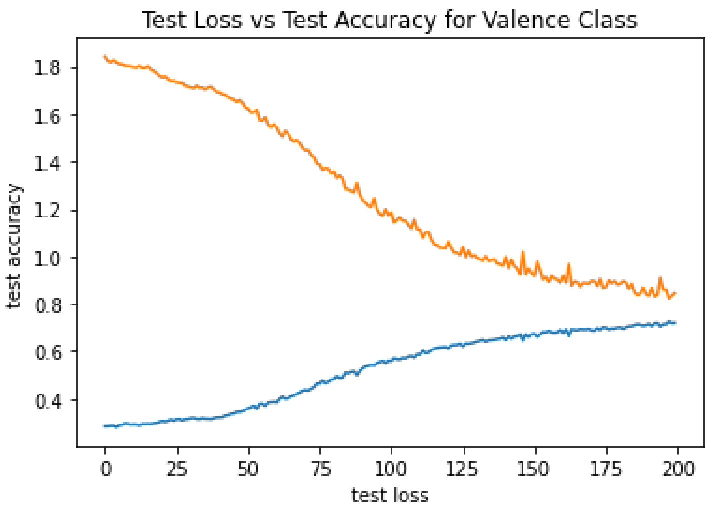

Figure 4.4: LSTM accuracy vs loss of valence class

ML models	Arousal%	Valence%	Dominance%	Liking%
KNeighborsRegressor	93	93	93.1	93.9
Bagging Regressor	79	81.7	81.6	83.5
Random Forest Regressor	78.8	81.9	81.6	83.6
SVR	64.7	66.2	67.8	69.4
AdaBoost	59.5	58.6	62.4	66.5

Table 4.2: Performance of our ML models on the 4 classes

borsRegressor with accuracy for all labels more than 90%. After that we have Bagging Regressor and Random Forest Regressor which performed also well with accuracy values more than 78% for all labels. But for SVR and AdaBoost, the result is very poor with accuracy values within 58%-70% range for all labels.

5 Conclusion

We have used EEG data collected from the state of the art DEAP dataset to analyze and detect the emotion of human beings. In our dataset, emotions were classified into - arousal, valence, dominance and liking. Long Short Term Memory (LSTM) and Convolution Neural Network (CNN) have been applied over the model. Both the LSTM and CNN models were hypertuned to suit the best parameters which give the best results. Based on our work, our CNN gave 87.6% accuracy on the "liking" class and our LSTM gave 87% accuracy also on the "liking class". We have applied our models to all the 4 classes of emotions given in the DEAP dataset, unlike other related works which only predicted two or three classes.

We also applied several Machine Learning techniques in our preprocessed dataset - KNR, Bagging Regressor, Random Forest Regressor, SVR, AdaBoost, of which, K-Nearest Neighbor (KNR) gave the best accuracy of 93.9% on the "liking" class and also could able to predict the other 3 classes with accuracies above 90%.

In the 2nd part of our thesis (CSE499B) we would like to extend this work by detecting emotion using face videos and also implementing the Facial Action Coding System. Then we would be merging all our pipelines to give an overall accuracy for both visual and EEG data to detect emotions.

References

[1] Samarth Tripathi, Shrinivasa G Acharya, Ranti Dev Sharma, Sudhanshu Mittal, and Samit Bhattacharya. Using deep and convolutional neural networks for accurate emotion classification on deap dataset. In *AAAI*, 2017.

[2] Dunne Campbell, R. A. Dunne, and N. A. Campbell. On the pairing of the softmax activation and cross–entropy penalty functions and the derivation of the softmax activation function, 1997.

[3] Peixiang Zhong, Di Wang, and Chunyan Miao. Eeg-based emotion recognition using regularized graph neural networks. *IEEE Transactions on Affective Computing*, 13(3):1290–1301, 2022.

[4] Yang Li, Lei Wang, Wenming Zheng, Yuan Zong, Lei Qi, Zhen Cui, Tong Zhang, and Tengfei Song. A novel bi-hemispheric discrepancy model for eeg emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, PP:1–1, 06 2020.

[5] Hao Tang, Wei Liu, Wei-Long Zheng, and Bao-Liang Lu. Multimodal emotion recognition using deep neural networks. pages 811–819, 10 2017.

[6] Divya Acharya, Shivani Goel, Harshit Bhardwaj, Aditi Sakalle, and Arpit Bhardwaj. 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)*, pages 1–8, 2020.

[7] Louis A. Schmidt and Laurel J. Trainor. Frontal brain electrical activity (eeg) distinguishes valence and intensity of musical emotions. *Cognition and Emotion*, 15(4):487–500, 2001.

[8] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7:162–175, 2015.

[9] Peng Chen and Jianhua Zhang. Performance comparison of machine learning algorithms for eeg-signal-based emotion recognition. In *International Conference on Artificial Neural Networks*, pages 208–216. Springer, 2017.

[10] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

[11] Salma Alhagry, Aly Aly Fahmy, and Reda A El-Khoribi. Emotion recognition based on eeg using lstm recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 8(10), 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, 9, 1996.

[13] Tormod Næs, Knut Kvaal, Tomas Isaksson, and Charles Miller. Artificial neural networks in multivariate calibration. *Journal of Near Infrared Spectroscopy*, 1(1):1–11, 1993.

[14]

[15] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.