International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

EEG Based Emotion Recognition Using Ensemble Models and Laplacian Eigenmaps

Megha D. Bengalur*[1], Jayachandran Arumugam[2], Roopesh G. Talikoti[3]

Submitted: 17/09/2023 **Revised**: 19/11/2023 **Accepted**: 30/11/2023

Abstract: Electroencephalography (EEG) signals provide valuable insights into human brain activity and emotional states. The DEAP dataset is a widely used resource containing EEG data and corresponding emotion labels recorded from participants watching music videos. Extracting meaningful EEG signal features is crucial for the recognition of emotions and other brain-computer interface applications. In this work, we explore the effectiveness of Laplacian Eigenmaps, a nonlinear dimensionality reduction technique, for extracting discriminative features from EEG signals on the DEAP dataset. We present an experimental study where we apply Laplacian Eigenmaps to reduce the high-dimensional EEG data into a lower-dimensional representation. Subsequently, we employ ensemble machine learning classifiers viz., Random Forest (RF) and eXtreme Gradient Boosting (XGB) classifiers for emotion classification based on the reduced features. The results demonstrate the capability of Laplacian Eigenmaps in capturing the underlying structure of EEG data, leading to improved emotion recognition accuracy with RF 98.1% and XGB 98.7% compared to other feature extraction.

Keywords*: Laplacian Eigenmaps, Dimensionality Reduction, Random Forest, eXtreme Gradient Boosting, Eigen values and Eigen Vectors, Ensemble Machine Learning, EEG Emotion Recognition.*

1. Introduction

Emotion recognition from ElectroEncephaloGram (EEG) signals is a challenging and significant area of research in affective computing and human-computer interaction. EEG-based emotion recognition aims to decode the emotional states of individuals by analyzing their brainwave patterns, which are directly related to their emotional experiences. The DEAP (Database for Emotion Analysis using Physiological Signals) dataset is one of the most used datasets for developing and evaluating EEG emotion identification models[1].

The DEAP dataset contains EEG, peripheral physiological signals, and subjective ratings collected from participants while they were exposed to audiovisual stimuli eliciting various emotional states. The EEG signals are non-invasive recordings of electrical activity in the brain captured using electrodes that are placed on the scalp. Each participant rated their emotional experiences based on valence (i.e positive to negative) and arousal (i.e calm to excited) dimensions after each stimulus presentation. The dataset is valuable for training machine learning models to recognize emotions from EEG signals.

Several studies have utilized the DEAP dataset to develop effective EEG emotion recognition algorithms. These studies often involve pre-processing the EEG signals, feature extraction, and employing machine learning or deep learning techniques to classify emotions based on the extracted features[2] [3] [4]. There are also studies which perform emotion analysis using Ensemble learning methods. [23]Veeramallu proposes EEG based emotion recognition automatically using Random Forest and EMD classifiers with accuracies of 89%, 91%, 93% for negative, positive emotions on SEED dataset.

The combination of the DEAP dataset and advanced EEG emotion recognition techniques holds promise for enhancing emotion-aware applications, such as affective computing systems, human-robot interaction, and virtual reality experiences. The ongoing research in this field continues to improve performance of classification and efficiency of emotion recognition from EEG signals, making it a captivating area with potential real-world applications.

The remaining of the paper is structured as follows. Section 2 deliberates work related to EEG signal classification. Section 3 demonstrates the feature extraction with dimensionality reduction technique, to provide an effective clarification for the EEG signal classification. Section 4 provides EEG signal classification with the model performance. Section 5 shows the experimental setup. Section 6 we analyze the achieved results of the proposed method following which we conclude the paper.

[1] *Computer Science & Engg.,, Presidency University, Bengaluru, India*
ORCID ID : 0000-0003-3201-694X
[2] *Computer Science & Engg.,, Presidency University, Bengaluru, India*
ORCID ID : NA
3 Software Developer, Maropost India Pvt. Ltd., Mohali, Punjab, India
ORCID ID : NA
[1]*meghabengalure@gmail.com*
[2]*ajayachandran@presidencyuniversity.in*
[3]*roopesh.talikoti@gmail.com*

2. Related Work

2.1. Feature Extraction And Emotion Analysis

While a wide range of domains is involved in emotion recognition studies, our focus in this study is on EEG-based analysis. EEG, being a measure of brain activity, is used to identify emotion states, as emotions originate in the brain [5]. Consequently, emotion recognition through EEG is an active area of research. Researchers mine various features from the EEG signals for emotion recognition, and the choice of the features is often based on speculative considerations. For instance, EEG asymmetry in left, right frontal regions has been associated with positive, negative emotions respectively [6], making EEG asymmetry a topic of great interest in emotion studies [7]. Additionally, variations in the power spectrum of different EEG bands serve as prime identifiers of emotional states [8]. However, there exists no consensus on the most appropriate features for emotion recognition [12]. Hence, a comprehensive evaluation of different feature types becomes crucial for developing emotion recognition systems. Although some studies have equated the prominence of various EEG features for emotion recognition [14], they either focused on the extraction of the similar feature characteristics using diverse computational methods or their comparisons were not extensive. Consequently, the probe for a robust feature set for realizing emotion identification systems remains limited, hindering the practical application of such systems in Human-Computer Interaction (HCI). To address this gap, our current work implements emotion recognition system based on EEG signals and evaluates the robustness of features chosen from most used feature types, namely entropy features [15], power features [13], fractal dimension features [19], wavelet features [15] and statistical features [12].

2.2. Emotion Models

Researchers use two main representations to model emotional states: discrete and dimensional models. In the discrete model, emotions are categorized according to the theory of basic emotions, which posits that all emotions are characterized by a primitive set of fundamental emotions [9]. However, the specific number and nature of these basic emotions are subjects of controversy and debate among researchers. Some argue that there exists exactly four basic emotions, namely disgust, anger, fear and happiness [10], while others propose a set of 27 different emotions [11]. Alternatively, the dimensional model represents emotions across multiple dimensions, such as arousal and valence. The well-known circumflex model is an example of the dimensional model, which is based on two dimensions: arousal, valance. Valence refers to the quality or how pleasant an emotion is, which ranges from low valence (unpleasant) to high valence (pleasant), while arousal indicates the level of excitement associated with the emotion, ranging from low arousal (calm) to high arousal(excited). The 2d model allows for a broader representation of emotions, but there is a possibility that it might lack the precision needed to fully capture the complexity of emotional experiences.

2.3. Dataset

The research work uses a DEAP dataset [https://www.eecs.qmul.ac.uk/mmv/datasets/deap/readme.html]. The study included 32 subjects who watched a set of 40, one-minute music video extracts. EEG signals were recorded from every participant during the study.

The study uses the pre-processed DEAP dataset of EEG recordings which has the EEG signals of 32 subjects [16]. The dataset is pre-processed which is downsampled, artifacts are removed, bandpass filtering is done from 4–45 Hz, segmentation and referencing etc. The dataset is open source and available for research scholars to perform the data analysis quickly with hassle free basic preprocessing. The process uses music videos as stimuli to evoke emotions. The participants watch a set of 40 60 seconds music video and their corresponding EEG signals are captured through the Biosemi ActiveTwo system. After every one minute video the participants documented their ratings using Likert scale which has ratings from 1-9 where 1 is low and 9 is high for arousal, dominance, liking, valence and familiarity level. In the proposed study we only consider the three-dimensions of emotions as arousal, valance and dominance. The self-assessment ratings of the participants for each of these three dimensions are used as labels. If the rating of the self-assessment is less than 5, the the corresponding label is considered low and if rating is same to or higher than 5, the respective emotion label is set to high. The participants face recordings are not used for analysis, hence the corresponding data related to it is discarded. Valence refers to pleasantness ranging from happy to unhappy. Arousal measures the emotion intensity from very excited to not excited. Dominance is the control on the emotions which ranges from total control to no control.

The study also makes the face video of the volunteers available, however, for this approach, we do not consider the same. The EEG recordings were available as 32 '.bdf' files as a 48-channel recording at 512 Hz. 22 of the volunteers were recorded in Twente and the remaining 10 at Geneva. The EEG channels followed the 10/20 naming system for locations and indices labeling. Figure 1 displays the dataset definitions.

Column name	Description
Online_id	The video id corresponding to the same column in the video_list file.
Valence	The valence rating (integer between 1 and 9).
Arousal	The arousal rating (integer between 1 and 9).
Dominance	The dominance rating (integer between 1 and 9).
Wheel_slice	The slice selected on the emotion wheel. For some participants the emotion wheel rating was not properly recorded. In these cases, the Wheel_slice value is 0. Otherwise, the mapping of emotions on the wheel to integers given here is: 1. Pride 5. Relief 9. Sadness 13. Envy 2. Elation 6. Hope 10. Fear 14. Disgust 3. Joy 7. Interest 11. Shame 15. Contempt 4. Satisfaction 8. Surprise 12. Guilt 16. Anger
Wheel_strength	The strength selected on the emotion wheel (integer between 0=weak and 4=strong).

Fig. 1. Describes the parameters of the self-study. Valence, Arousal and Dominance are discretized between 1 and 9. The Wheel Slice lists 16 emotions that the volunteer can feel and the Wheel Strength describes the strength of the emotion that was felt by the volunteer

3. Methods

In the proposed research work we have implemented a ensemble machine learning model to identify emotions based on EEG. Emotions are categorised into three-dimensions as arousal, valence and dominance. The sections below introduces a detailed explainition of the fetaure extraction, classification methods. Figure 2 displays the proposed work in block diagram.

3.1. Preprocessing of EEG Data

The raw EEG data is preprocessed, including filtering to remove noise and artifacts, resampling if necessary, and segmenting the data into smaller epochs around emotion-related events or stimuli.

3.2. Feature Extraction

The EEG data captured for the problem comes with a huge set of features which involves readings from 32 channels where each channel has time domain data recorded for 63 seconds at 128Hz. So, in total the channel readings maps to 128Hz * 63 seconds = 8064 reading per channel. This applies to all the available channels. Along with this there exists the ratings data generated by the participants for every video they have watched. Thus, creating a huge dataset.

There exists a number of feature extraction techniques that have already been explored, they include features like frequency domain, time domain, time-frequency domain, statistical, fractal dimensions, event related potential and many others.

Alongside, dimensionality reduction can indeed be considered as a feature extraction technique for EEG emotion recognition. As mentioned earlier, EEG data is high-dimensional and non-linear, and reducing the number of features can enhance the performance of emotion recognition models and decrease computational complexity. The most commonly used dimensionality techniques techniques are Principal Component Analysis (PCA), Independent Component Analysis (ICA). In this paper, we investigate the utility of Laplacian Eigenmaps for the reduction of dimensionality and feature extraction on the DEAP dataset.

3.2.1. Laplacian Eigenmaps

Laplacian Eigenmaps(LE) [24] is a manifold learning technique that targets to preserve the local structures in the data space when projecting it to a lower-dimensional subspace. It constructs a similarity dataset based on pairwise distances between data points and computes the graph Laplacian dataset. By solving the eigenvalue problem of the Laplacian dataset, Laplacian Eigenmaps derives the lower-dimensional representations of data. The block in Figure 1. displays the steps involved in the Laplacian Eigenmaps dimensionality reduction technique. The following section explains in detail about the steps involved in the LE with the pseudocode.

Pseudocode for Laplacian EIgenmaps

Input: Data dataset of size (n x m), where n represents the total number of data values, and m is the original dimension of individual data point.

k: Desired lower-dimensional embedding dimension.

Output: Dataset of size (n x k) representing the lower-dimensional embedding of the data points.

Step 1: Construct Similarity Dataset (W)

```
func AffinityDataset(Y)
  for i ϵ 1……..n do
      for  j ϵ 1…… n do
         Compute the similarity between data points Y[i] and Y[j]
         W[i][j] <= similarity(Y[i], Y[j])
      end for
  end for
```

Step 2: Compute Degree Dataset (D)

```
Func DegreeDataset(W)
  for i ϵ 1 …… n do
      for  j ϵ 1 …… n
            D[i][i] <- Σ W[i][j]
      end for
end for
```

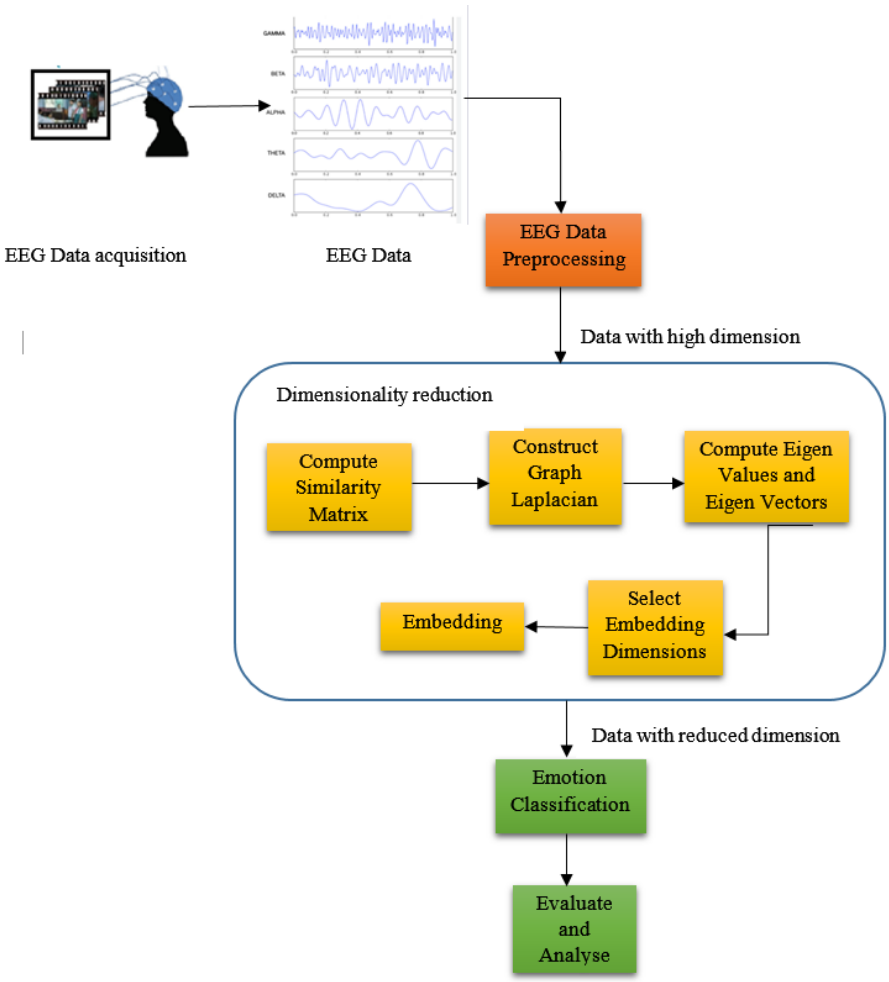


Fig. 1. Proposed Methodology

Step 3: Compute Graph Laplacian (L)

Func Laplacian(W, D)

L <- D - W

Step 4: Eigenvalue Decomposition of L

Compute the eigenvectors V and eigenvalues λ of L

Step 5: Select Top k Eigenvectors

Sort the eigenvalues λ and their corresponding eigenvectors V in ascending order based on the eigenvalues' magnitudes. Select the k eigen vectors which corresponds to the smallest k non-zero eigenvalues and form a dataset V_k of size (n x k) with these k eigenvectors.

Step 6: Embedding the Data

The dataset V_k represents the low dimensional embedding of the data values in a new space.

Set Y = V_k

Step 7: Output

Return the dataset Y of size (n x k) as the resulting low dimensional representation of the input data values.

The resulting dataset Y is the lower-dimensional representation of the input data points, capturing the intrinsic structure and relationships in a lower-dimensional space. This representation can be used for visualization, clustering, or as input to other machine learning models for various downstream tasks.

Each of the above steps in the pseudocode is explained in detail below:

1) Similarity Dataset: In data analysis and machine learning, similarity matrices are often used to represent pairwise similarities between data points or features. For calculating the pairwise similarity we first obtain a distance dataset using Euclidean distance and then generate a similarity dataset using a similarith measure. The formula for Euclidean distance dataset is given below:

$$dist(x, z) = \sqrt{((x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots\ldots + (x_n - z_n)^2)} \quad (1)$$

To construct a similarity dataset, we first need to define a similarity measure, which is a function that maps pairs of datapoints to a numerical value between 0 and 1, where 0 represents complete dissimilarity and 1 represents complete similarity.

From the above dataset we calculate the similarity dataset as follows:

$$\text{Similarity Measure} = \frac{1}{1-\text{dist}(x,\ z)} \quad (2)$$

Once we have a similarity measure, we can calculate the similarity between all pairs of objects and store the results in a dataset. The resulting similarity dataset is usually symmetric, with 1s along the diagonal representing the similarity of each object with itself.

2) Compute the Degree dataset: From the affinity dataset, we compute the degree dataset D. It is a dataset where the i-th diagonal element Diag[i][i] corresponds to the summation of the ith row of the similarity dataset W. The degree dataset D is an n x n diagonal dataset, where D[i][i] represents the degree (the sum of edge weights) of node i. It is computed using (3).

$$\text{Diag}[i][i] = \Sigma W[i][j] \text{ for all j from 1 to n} \quad (3)$$

3) Construct the Graph Laplacian: The graph Laplacian encodes information about the connectivity and local smoothness of the data points within the graph. It measures how each data point is related to its neighboring data values in the high-dimensional space. This information is crucial in understanding the underlying structure and geometry of the data, especially when the data resides in a nonlinear manifold like in the signals of EEG data. More specifically, the graph Laplacian explains:

The unnormalised Laplacian dataset L is calculated as the difference between the degree dataset D and the similarity dataset W:

$$L = D - W \quad (4)$$

To convert from the unnormalized to normalized Laplacian, we can do:

$$L\_normalized = \frac{1}{\sqrt{D}} * L * \frac{1}{\sqrt{D}} \quad (5)$$

Normalization makes nodes with different degrees have roughly equal influence, thus, balancing the influence of nodes. It can improve clustering and classification. Since node influence is balanced, the normalized Laplacian can give better results for the algorithms that uses it for tasks like spectral clustering and graph-based classification.

4) Eigenvalue Decomposition: Eigenvalue decomposition is a fundamental dataset factorization technique that decomposes a dataset into a set of eigenvalues and eigenvectors. The significance of eigenvalue decomposition in Laplacian Eigenmaps lies in its ability to extract essential information from the graph Laplacian dataset (L). The dataset consists of multiple complex eigenvalues, so we choose QR Eigenvalue decomposition approach. The pseudocode for the same is described below:

Pseudocode for QR Decomposition:

```
Input: Square matrix L_Normalized, Number of iterations N
Function qr_eigenvalue_decomposition(A, N):
  Initialize matrix A_k = A
  Initialize matrix Q = I (identity matrix)
  for i = 1 to N do:
    {
      Q, R = QR_decomposition(A_k)
      A_k = R * Q
    }
  eigenvalues = diagonal_entries(A_k)
  eigenvectors = []
  for each eigenvalue in eigenvalues do:
   {
    Solve (A - eigenvalue * I) * x = 0 for eigenvector x using any appropriate method
    Normalize x to obtain the eigenvector
    Append eigenvector to eigenvectors
    }
  Return eigenvalues, eigenvectors
End Function
```

Spectral Dimensionality Reduction: By performing eigenvalue decomposition on L, we obtain its eigenvalues (λ) and corresponding eigenvectors (V). The eigenvectors represent the low-dimensional coordinates of the data values in a new space, and the eigenvalues indicate their importance or variance.

5) Selecting embedding dimensions into Lower-dimensional Space: The k smallest non-zero eigenvalues and their corresponding eigenvectors are selected. These eigenvectors form a new dataset V_k of size (n x k), where n represents the number of data values and k is the desired lower-dimensional embedding dimension. The dataset V_k represents the lower-dimensional embedding of the data points.

Overall, eigenvalue decomposition is a key step in Laplacian Eigenmaps, as it allows for non-linear reduction of dimensionality, capturing intrinsic structure of the data, and facilitating visualization and clustering tasks in a lower-dimensional space. Finally, this representation we pass to the classifier.

3.3. Ensemble Learning Algorithms for the proposed work

While selection of an efficient machine learning algorithms to recognize emotions is a challenging task, there has been many studies in the past employing the ML algorithms to identify emotions. Ensemble learning algorithms, like Random Forest (RF) , eXtreme Gradient Boosting (XGB), shows exceptional accuracy in the classification of emotions using EEG data [18] and [19] respectively . These algorithms are able to achieve high accuracy by combining the predictions of multiple base learners, which helps to mitigate the emotion classification[17]. In our research study we examine the performance of the these two ensemble learning algorithms on our extracted feature set for emotion classification.

3.3.1. Random Forest

Random Forest(RF) is an ensemble decision tree-based learning algorithm that is used for classification in our study. It constructs multiple decision trees during the training phase and then combines their predictions to make predictions. RF uses a technique called bootstrapping where it randomly samples the training data using replacement to create multiple subsets called "bootstrap samples". Each subset is used to train an individual decision tree.

When creating the decision trees, RF introduces randomness by choosing features that are random subsets created from the dataset for making split decisions at each node of the tree. This helps to reduce overfitting and increases the diversity among the individual trees.

The decision trees in the RF are built using the bootstrap sample and the random subset of features. The trees are constructed using the standard decision tree algorithms CART, where nodes are split based on the selected feature that provides the best separation. Each decision tree in the RF independently classifies a data point based on its majority class prediction. The majority class from the individual decision trees is used to predict the final class. The improvement of the algorithms' performance is achieved by tuning the hyperparametres.

The working principle of RF algorithm:

For each decision tree, an i^{th} sample from the bootstrap i.e. BS(i) from BS is selected. Features from random subsets are choosen at every node of the tree despite taking into account all the feature-split possibilities i.e. f $\underline{C}$ F, where F is the feature set. Each node splits with the optimal feature in f. The pseudo-code is presented below.

Pseudocode of Random Forest:

Dataset BS = { (x_1, y_1) …. (x_k, y_k) }, where 'F' is the feature set and 'K' is the number of trees in the forest.

```
function RF (SB , F)
{
        T ← { }
        for n=1… K do
                {
                 BS(n) ← A bootstrap data sample
                          choosen from set S
                 t n ← RandomTree(BS(n), F)
                 T ← T ∪ { t n }
                }
        return T
}
function RandomTree (BS , F)
{
        At every node of the tree:
                for each subset of F
                        {
                        f ←split based on best
                            feature
                        }
                return (tree built)
}
```

3.3.2. eXtreme Gradient Boosting

eXtreme Gradient Boosting(XGB) is an ensemble learning method which builds a robust predictive model by joining the predictions of many weak models, basically the decision trees. Decision trees are used as the base or weak learners by XGB. We refer to these treesto as "stumps" because they are typically very shallow trees with only a few nodes. Each decision tree is a simple model that makes binary decisions at each node. XGB is based on the gradient boosting framework. In gradient boosting, models are built sequentially, and each new model focuses on correcting the errors occurred in the previous models.

The procedure begins with an initial tree, often a simple one like a uniform distribution for classification tasks. It calculates the residuals (actual and predicted values difference) for each data point. These residuals represent the errors made by the current model. A new decision tree is then fit to predict these residuals. The new tree's predictions are weighted and added to the previous model's predictions. The weights are determined using a technique called Gradient Descent, which minimizes the log loss function.The process is repeated quantified number of iterations (trees) or til a predefined terminating condition is reached. XGB incorporates regularization techniques to prevent overfitting. In our proposed work we have used a L1 (Lasso) regularization to penalize large coefficients in the decision trees. The main reason for choosing XGB is it performs the

processing parallely leading to faster training. The missing vaues in the dataset is handled very well, which can help in finding out which features are the most influential in making predictions. The model also comes up with a good predictive power and provides a better accuracy than other models.

Mathematically, the algorithm can be explained as follows:

(1) Let us assume that a model has n decision tree, then the models classification can be represented by:

$$\hat{y}_i = \sum_{n=1}^{N} c_n(x_i), c_n \in C \quad (6)$$

where, n represents the no. of trees, c is the functional space of C, C is the set of possible classifiers.

(2) The objective function for the above model is given by:

$$\text{obj}\ (\theta) = \sum_{i=1}^{n}\ l(y_i, \hat{y}_i) + \sum_{n=1}^{N}\ \Omega(c_k) \quad (7)$$

where, the first summation represents loss function and the second summation represents the regularization parameter. The function $\Omega(c_k)$ is a regularization parameter to avoid overfitting defined as:

$$\Omega(c) = \gamma P + \frac{1}{2}\lambda \sum_{j=1}^{P} w_j^2$$

Where P represents the number of leaf nodes and w_j is the weight score on the jth leaf node.

(3) The second order taylor series expansion derivative is given by:

$$obj(\theta) = \sum_{j=1}^{n}\ \left[l\left(y_j, \hat{y}_j^{(\theta-1)}\right) + g_j c_t(\text{xj}) + \frac{1}{2} h_j c_t^2(x_j)\right] + \Omega(c_t) \quad (8)$$

where g_j and h_j can be defined as:

$$g_j = \partial_{\hat{y}_j^{(t-1)}}\ l\left(y_j, \hat{y}_j^{(t-1)}\right)$$

$$h_j = \partial^2_{\hat{y}_j^{(t-1)}} l\left(y_j, \quad \hat{y}_j^{(t-1)}\right)$$

(4) Rewriting (9) by substituting the equation for Ω, the objective function becomes:

$$obj(t) \approx \sum_{j=1}^{P}\ \left[g_j c_t(\text{xj}) + \frac{1}{2} h_j c_t^2(x_j)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{P}\ w_j^2 \quad (9)$$

The above expression can be simplified to:

$$obj^{(t)} = \sum_{j=1}^{P}\ \left[G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2\right] + \gamma P \quad (10)$$

Where,

$$G_j = \sum_{i \in I_j} g_i$$

$$H_j = \sum_{i \in I_j} h_i$$

wj are the leaf scores given by:

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

The objective function can be optimized to get the best fit as follows:

$$\text{obj}^* = -\frac{1}{2}\sum_{j=1}^{P} \frac{G_j^2}{H_j + \lambda} + \gamma \quad (11)$$

Now, we measure the efficiency of the tree(how good the tree is)is, we cannot optimize the entire tree directly, so we try to optimize individual nodes one at a time. More specifically we try to split a leaf into two leaves (left and right leaf). The score of this split is gain given by:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{\lambda + H_L} + \frac{G_R^2}{\lambda + H_R} - \frac{(G_R + G_L)^2}{H_L + \lambda + H_R}\right] - \gamma \quad (12)$$

Finally to best tree construction, the splitting point is repeatedly calculated till we reach the highest depth. The nodes having gain lesser than zero are then pruned in the bottom-up approach.

4. Performance Evaluation

The accuracy of the machine learning model is calculated using the performance metrics. For our classification models we determine the classification summary containing Accuracy, Presicion, Recall, F1 score(Table 2) generated with the help of Confusion matrix. Confusion matrix is a tabular representation(Table 1) that provides a summary of the model's decision making compared to the actual true values in the dataset. The matrix with its components terms are given below:

TABLE 1: CONFUSION MATRIX

	Predicted	
Actual	***Positive***	***Negative***
Positive	TP	FN
Negative	FP	TN

Where,

True Positives (TP): Model predicts positive observations as positive.

True Negatives (TN): Model predicts negative observations as negative

False Positives (FP): Model predicts negative observations as positive.

False Negatives (FN): Model predicts positive observations as negative.

Using the above matrix the other performance metrics are determined as given below:

TABLE 2. PERFORMANCE METRICS

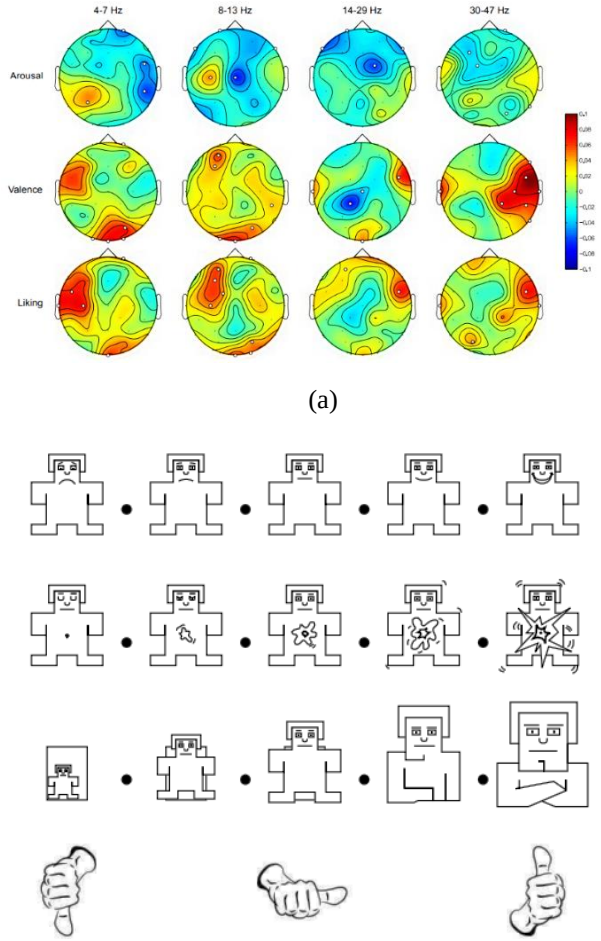
Sl. No.	*Metrics*	*Formula*
1	Accuracy	$\frac{TN + TP}{TN + TP + FN + FP}$
2	Precision	$\frac{TP}{TP + FP}$
3	Recall	$\frac{TP}{TP + FN}$
4	F1 Score	$2 * \frac{Precision * Recall}{Precision + Recall}$

5. Experimental Setup

The experminetal setup of the DEAP dataset for the study is shown in figure 3. The figure displays a person with a head mounted EEG cap(Biosemi ActiveTwo system) to record EEG and peripheral physiological signals with dedicated recording PC sitting in a laboratory environment with controlled illumination. A softare developed by Neurobehavioral systems is used to showcast the stimuli and record the ratings given by the user. A 17-inch screen (1280 × 1024, 60 Hz) is used to present the music video and to reduce the subject's eye movements, the video stimuli are displayed at a resolution of 800 × 600. A 1 meter disance was maintained between the subject and the video stimuli. A relative loud volume music with Stereo Philips speakers were used. The participants are made to wear the electrode cap consisting of 32 electrodes that are mounted according to10 – 20 International System and the signals are recorded at a sampling rate is 512 Hz.

Fig. 3. Participant before the EEG recording process

After the completion of each trial, subjects were informed to performe a self assessment to determine their liking, arousal, dominance and valence levels. Self Assessment Manikin (SAM) [20](figure 4(b)) were displayed to visualize the scenes and understand the scales for each emotions. The manikins were shown in the screen centre with the rating from 1 to 9 displayed below. To indicate the participants self-assessment level they dragged the mouse horizontally across the numbers. The valence continuum spans from feelings of unhappiness or sadness to feelings of happiness or joy. The arousal spectrum extends from states of calmness or boredom to states of heightened stimulation or excitement. The dominance gradient spans from being submissive or feeling without control to being dominant or feeling in control and empowered. Participants' individual preferences and likings regarding the video is indicated by the fourth scale.



(a)

(b)

Fig. 4. (a) The broad frequency bands gmma(30-47Hz), beta(14-29Hz), alpha(8-13Hz), theta(4-7Hz), for the Arousal, Valance and Liking emotions averaged over all the participants. (b) Self Assessment Manikins from the top: valance, Arousal, Dominance, Liking.

6. Results and Discussions

The EEG data is taken from DEAP dataset for three emotional states. The particulars of the emotions identified in the proposed work is defined in fig.4(a & b). The figure displays the different emotion types recognized and along with the broad frequency bands gamma(30-47Hz), beta(14-29Hz), alpha(8-13Hz), theta(4-7Hz) pertaining to different emotional states as Valance, arousal, dominance.

Once the EEG signals data is preprocessed, thirty two features were extracted from thirty two different electrodes

attached on each subject scalp in time–frequency domain. The frequency domain data is then converted to time domain usin Fast Fourier transform. This time domain data is then forwarded to Laplacian Eigenmaps for dimensionality reductions. The eigen vectors obtained from LE is treated as the reduce feature space. The reduced features set are fed to the ensemble machine learning models RF and XGB as inputs for identifying the emotional states. We use the Sklearn[22] library to implement the ensemble learning models i.e RF and XGB, available in Python's ML package.

Hyperparametre Tuning of the Ensemble Models: Machine learning algorithms come with external parameters called as hyperparameters, which play a vital role in controlling the learning process and determining the optimal configuration of the model's parameters upon completion of training. Tuning these hyperparameters is essential for enhancing the trained model's performance.

In this study, we conducted hyperparameter tuning for Random Forest (RF) and XGBoosting (XGB) using the 'GridSearchCV' API provided by Scikit-learn. K-fold cross-validation was employed for each individual dataset to ensure robustness. For both RF and XGB classifiers, we explored a range of parameter values and applied 5-fold cross-validation to improve the reliability of our results. Specifically, we focused on tuning two key hyperparameters of XGB: 'n_estimators' and 'max_depth'.

The hyper parameters used for the proposed ensemble models for EEG based emotion recognition with GridSearchCV are listed in Table 3.

TABLE 3. Classifier parametre settings

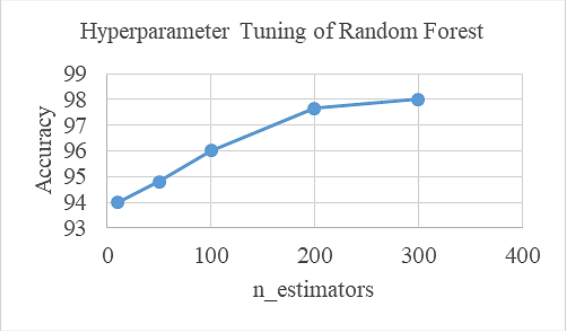
Ensemble Learning Classifier	***Parametre settings***
Random Forest	random_state = 42, base_estimator = _DecisionTreeClassifier, n_estimators = 400, max_features = 'auto', criterion = 'gini'
eXtreme Gradient Boosting	base_estimator = _DecisionTreeClassifier, n_estimators = 280, max_features = 'auto', criterion = 'gini'

The 'max_depth' hyperparameter determines the extreme depth of the decision tree, helping to control overfitting, while 'n_estimators' represents the total no. of parallel threads used in the algorithm execution. Our analysis, as illustrated in Figure 5, revealed that GridSearchCV identified the maximum values for these hyperparameters as 'n_estimators' = 300, 'max_depth' = 7 w.r.t accuracy being considered the scoring metric.

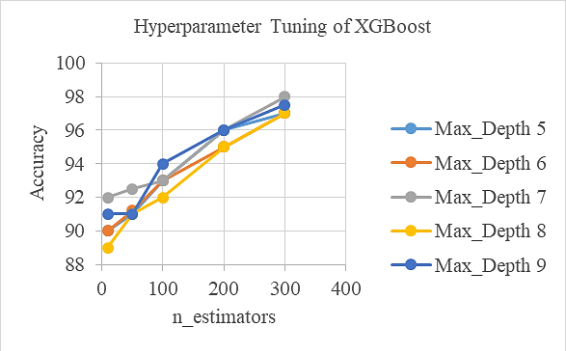
Fig. 5 (a) & (b) showcases the accuracy of RF and XGB algorihms with their hyper-parameter being tuning viz., Max_depth and n_estimators. The best value of n_estimator for both the approaches is found to be 400 and 280 as seen in the plot

We trained the RF model using a 5-fold cross validation. Out of a total of 1280 samples, 1198 were correctly classified, yielding an accuracy of 98.17%. This high accuracy demonstrates the RF classifier's effectiveness in emotion classification for EEG data. Similarly, the XGBoosting classifier achieved 98.78% accuracy by correctly classifying 1209 out of 1280 samples. To evaluate these algorithms' performance, we compared them to the K-nearest neighbour classifier [25] [26] [27].

A comprehensive performance analysis of the proposed classifiers and the KNN classifier is provided in Table 4. This analysis includes metrics such as recall, emotion classification accuracy, F1-score, precision, for 3 different emotion class.



(a)



(b)

Fig. 5. (a) Hyperparameter (n_estimator) tuning for RF (b) Hyperparameter (n_estimator & max_depth) tuning for XGBoosting.

7. Conclusion

In this work, we have demonstrated a novel approach for the recognition of emotion using physiological signals with results for the same. Laplacian Eigenmaps along with Ensemble Machine Learning Techniques were applied for feature extraction and classification respectively to recognise the human emotions. From the results achieved, it is

observed that both the ensemble methods i.e RF and XGB present a promising result in terms of the performance evaluation metrics. Random Forest achieves an accuracy of 98% and XGB with 98% on an average for all the emotions considered. The study showcases that, the matrix reduction technique(i.e LE) helps to reduce the feature space to a considerable extent. The future enhancements of the study can be to achieve greater feature extraction by combining the matrix transformations and the statistical approaches.

TABLE 4. Performance metrics of the proposed approach

Sl. No.	*Emotion*	*Classifier*	*Accuracy %*	*Precision %*	*Recall %*	*F1 Score %*
1.	Valance	Random Forest	98.31	97.92	98.11	97
		eXtereme Gradient Boosting	98.62	98.48	98.63	98.33
2.	Arousal	Random Forest	97.89	97.62	98.11	98.23
		eXtereme Gradient Boosting	98.5	98.77	98.1	98
3.	Dominance	Random Forest	97.65	97	97.60	97.80
		eXtereme Gradient Boosting	98.63	98.11	98.34	98.76
4	All	K-Nearest Neighbor	94%(overall)			
5.	All	SVM	96%(overall)			

References

[1] Koelstra, S., Muhl, C., Soleymani, M., Lee, J. S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., & Patras, I. (2012). DEAP: A Database for Emotion Analysis using Physiological Signals. IEEE Transactions on Affective Computing, 3(1), 18-31. DOI: [10.1109/T-AFFC.2011.15](https://doi.org/10.1109/T-AFFC.2011.15)

[2] Soleymani, M., Lichtenauer, J., Pun, T., & Pantic, M. (2012). A Multimodal Database for Affective Computing Research. In Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction (ACII), 791-796. DOI: [10.1109/ACII.2011.97](https://doi.org/10.1109/ACII.2011.97)

[3] Yang, Y., Zhang, Y., & Zheng, W. (2018). EEG-Based Emotion Recognition using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. IEEE Transactions on Affective Computing, 9(3), 321-334. DOI: [10.1109/TAFFC.2017.2665668](https://doi.org/10.1109/TAFFC.2017.2665668)

[4] Zheng, W., Lu, Y., & Yang, Y. (2015). Investigating Critical Frequency Bands and Channels for EEG-based Emotion Recognition with Deep Neural Networks. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI), 109-116. DOI:[10.1145/2818346.2820775](https://doi.org/10.1145/2818346.2820775)

[5] Bhatti AM, Majid M, Anwar SM, Khan B. Human emotion recognition and analysis in response to audio music using brain signals. Comput Human Behav 2016;65:267–75.

[6] Harmon-Jones E, Gable PA, Peterson CK. The role of asymmetric frontal cortical activity in emotion-related phenomena: a review and update. Biol Psychol 2010;84:451–62.

[7] Pizzagalli D, Regard M, Lehmann D. Rapid emotional face processing in the human right and left brain hemispheres: an ERP study. Neuroreport 1999;10:2691–8.

[8] Lan Z, Sourina O, Wang L, Scherer R, Müller-Putz G. Unsupervised feature learning for EEG-based emotion recognition. 2017 Int. Conf. Cyberworlds. 2017. pp. 182–5.

[9] Liu Y, Sourina O, Nguyen MK. Real-time EEG-based emotion recognition and its applications. Transaction on computational science XII. Springer; 2011. p. 256–77.

[10] Tuomas E, Vuoskoski K. A review of music and emotion studies: approaches, emotion models, and stimuli. Music Percept An Interdiscip J 2013;30:307–40. http://dx.doi.org/10.1525/jams.2009.62.1.145 b i o c y b e r n e t i c s a n d b i o m e d i c a l e n g i n e e r i n g 4 0 (2 0 2 0) 9 1 0 – 9 2 6924

[11] Cowen AS, Keltner D. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. Proc Natl Acad Sci U S A 2017;114:E7900–9.

[12] Jenke R, Peer A, Buss M. Feature extraction and selection for emotion recognition from EEG. IEEE Trans Affect Comput 2014;5:327–39.

[13] Thammasan N, Moriyama K, Fukui K, Numao M. Continuous music-emotion recognition based on electroencephalogram. IEICE Trans Inf Syst 2016;99:1234–41.

[14] Schaaff K, Schultz T. Towards emotion recognition from electroencephalographic signals. Affect. Comput. Intell. Interact. Work. 2009. ACII 2009. 3rd Int. Conf.. 2009. pp. 1–6.

[15] Ianez E, Azorin JM, Ubeda A, Fernandez E, Sirvent JL (2010) LDA-based classifiers for a mental tasks-basezd brain–computer interface. In: Proceeding of the 2010 IEEE international conference on systems man and cybernetics (SMC 2010), 10–13 Oct 2010. IEEE Press, pp 546–551.

[16] Koelstra S, Muhl C, Soleymani M, Lee J-S, Yazdani A, Ebrahimi T, et al. Deap: a database for emotion analysis; using physiological signals. IEEE Trans Affect Comput 2012;3:18–31.

[17] Parmar U, Pandya DH (2021) Comparison of the supervised machine learning techniques using WPT for the fault diagnosis of cylindrical roller bearing. Int J Eng Sci Technol 13(2):50–56

[18] G. K. P. Veeramallu, Y. Anupalli, S. k. Jilumudi and A. Bhattacharyya, "EEG based automatic emotion recognition using EMD and Random forest classifier," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944903.

[19] Parui, S., Bajiya, A. K. R., Samanta, D., & Chakravorty, N. (2019, December). Emotion recognition from EEG signal using XGBoost algorithm. In *2019 IEEE 16th India Council International Conference (INDICON)* (pp. 1-4). IEEE.

[20] J. D. Morris, "SAM: the self-assessment manikin. An efficient cross-cultural measurement of emotional response," Journal of Advertising Research, vol. 35, no. 8, pp. 63–68, 1995

[21] Available at https://engineering.case.edu/bearingdatacenter. Accessed 16 Sep 2021 Available at https://scikit-learn.org/stable/. Accessed 16 Sep 2021

[22] G. K. P. Veeramallu, Y. Anupalli, S. k. Jilumudi and A. Bhattacharyya, "EEG based automatic emotion recognition using EMD and Random forest classifier," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944903.

[23] Belkin, Mikhail, and Partha Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation." *Neural computation* 15.6 (2003): 1373-1396.

[24] Fan, J.; Wade, J.W.; Key, A.P.; Warren, Z.E.; Sarkar, N. EEG-Based Affect and Workload Recognition in a Virtual Driving Environment for ASD Intervention. IEEE Trans. Biomed. Eng. 2018, 65, 43–51. [CrossRef]

[25] Soroush, M.Z.; Maghooli, K.; Setarehdan, S.K.; Nasrabadi, A.M. Emotion Classification through Nonlinear EEG Analysis Using Machine Learning Methods. Int. Clin. Neurosci. J. 2018, 5, 135–149

[26] Li, Y.; Zheng, W.; Zong, Y.; Cui, Z.; Zhang, T.; Zhou, X. A Bi-Hemisphere Domain Adversarial Neural Network Model for EEG Emotion Recognition. IEEE Trans. Affect. Comput. 2021, 12, 494–504