Expert Systems With Applications 207 (2022) 118025

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

EEG-based emotion analysis using non-linear features and ensemble learning approaches

Md. Mustafizur Rahman [a,*], Ajay Krishno Sarkar [b], Md. Amzad Hossain [a,*], Mohammad Ali Moni [c]

[a] Department of Electrical and Electronic Engineering, Jashore University of Science and Technology, Jashore 7408, Bangladesh
[b] Department of Electrical and Electronic Engineering, Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh
[c] Artificial Intelligence & Digital Health Data Science, School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland, St Lucia, QLD 4072, Australia

ARTICLE INFO

Keywords:
Electroencephalography
Emotion recognition
Ensemble machine learning
Non-linear features
RFE

ABSTRACT

Recognition of emotions based on electroencephalography (EEG) has become one of the most emerging topics for healthcare, education system, knowledge sharing, gaming, and many other fields in the last few decades. This paper proposes three non-linear features (Higuchi fractal dimension, sample entropy, and permutation entropy) and eight ensemble learning approaches to predict six basic emotions (hope, interest, excitement, shame, fear, and sad). To increase the recognition rate of each classifier, we utilized a randomized grid search technique for tunning the hyperparameter of each algorithm. Moreover, the impact of electrodes on each emotion was observed using the recursive feature elimination (RFE) method. In addition, the synthetic minority oversampling technique (SMOTE) is used to handle the imbalanced sample distribution of each emotion. Then, we have conducted all the experiments on DEAP and AMIGOS datasets and calculated the computation time and accuracy to assess each algorithm's performance. Besides, we observe statistical significance to compare the algorithm's performance. From the experimental result, we achieved the highest average accuracy, 89.38% and 94.62%, using Higuchi fractal dimension on DEAP and AMIGOS datasets, respectively. We also observed that the Higuchi fractal dimension outperforms the sample entropy and permutation entropy in terms of accuracy. Our proposed method increased the average recognition rate by 8.22% and 1.77%, respectively, compared with existing approaches working on the same dataset. Along with accuracy, precision, recall, F-measure, and AUC-ROC have been evaluated using the same experimental setting on DEAP and AMIGOS datasets. The results of each classifier have been shown in the table and graph. Moreover, the proposed method outperforms existing approaches dealing with shallow machine learning techniques.

1. Introduction

Nowadays, human–computer interaction (HCI) systems are involved in every corner of our lives. The massive development of HCI technology and its uses in various fields have aroused great interest in developing more intelligent HCI. Human emotions undoubtedly play an essential role in building humanized human–machine interfaces in the field of HCI. Therefore, affective computing is an essential field of study that aims to build a model to identify emotions accurately (Picard, 2000). Much progress has been made by analyzing emotions in human–computer interaction in areas such as decision-making, healthcare for mentally disordered patients, and knowledge sharing using eye-tracking. Moreover, it is also possible to develop an HCI-based cloud-assisted platform to improve healthcare services and education (Hossain, 2017). Emotions are human response that reflects the personal significance of a thing, an event, or a state of affairs. According to David G. Myers, human emotion implies physiological arousal, expressive behavior, and conscious experience (Myers, 2003). In 1972, Paul Ekman suggested six basic emotions: anger, fear, surprise, sadness, and disgust, all of which are associated with different facial expressions (Ekman, 1972). Another psychologist Robert Plutchik described eight primary emotions: anger, anticipation, joy, trust, fear, surprise, sadness, and disgust that are called Plutchik's wheel of emotions (Plutchik, 1980). This method explains that emotions might be merged to generate new

* Corresponding authors.
E-mail addresses: mustafizur.170710@gmail.com (Md.M. Rahman), mahossain.eee@gmail.com (Md.A. Hossain), m.moni@uq.edu.au (M.A. Moni).

https://doi.org/10.1016/j.eswa.2022.118025
Received 13 November 2021; Received in revised form 4 June 2022; Accepted 29 June 2022
Available online 2 July 2022

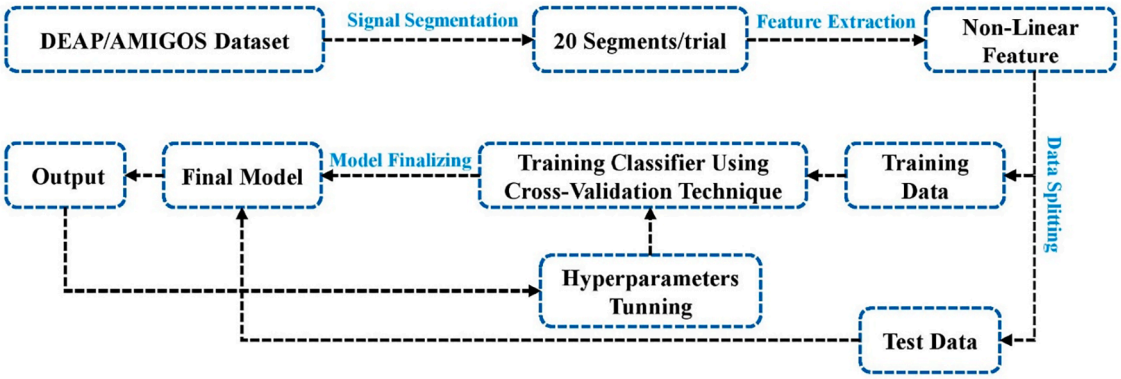


Fig. 1. Emotion recognition process based on our proposed method.

emotions in the same way that colors can be mixed to create new colors. In addition, emotions can be classified as positive, negative, and neutral based on valence and arousal level. Moreover, researchers have developed emotional models such as the valence-arousal-dominance model to represent emotions by considering specific brain regions that reflect emotions like positive and negative ones.

Russell (Russell, 1980) proposed a two-dimensional emotion model where emotions are described according to the valence and arousal plane. Valence describes positive or negative affectivity, while arousal measures how calming or exciting the information is. Later Mehrabian and Russel proposed another model called the valence, arousal, and dominance (VAD) model or three-dimensional models of emotion. In this model, they described pleasure, arousal and dominance are the deciding factors in defining emotional states adequately (Mehrabian & Russell, 1977). Therefore, any basic emotions can be extracted with the help of finding the value of valence, arousal and dominance level. Numerous researches implemented physiological signals, voice, body gestures, and facial expressions to distinguish emotions. Among them, electroencephalograph (EEG) offers meaning-rich signals with a high temporal resolution accessible using cheap, portable EEG devices. Other physiological signals (electrocardiograph, MEG, electromyography, skin temperature, and galvanic skin response) except electroencephalograph performed well, but those are capable only of finding specific emotions. For example, Arousal levels can be detected using ST and GSR, while valence is measured using EMG. Negative emotions (panic, fear, and sadness) can be detected using the Respiration Rate and ECG, but it can be problematic in the case of positive emotions (Rahman et al., 2021). Compared to behavioral responses (voice, body gestures, and facial expressions), EEG can easily identify the human's real emotions because the variation of EEG signals reflects changes in emotions. In addition, identification of emotions from EEG signal achieves higher accuracy as compared to other physiological signals. Different emotions have quite different patterns of neuronal activity as well as some brain regions play an important role in emotions analysis. Mohammadi et al. found positive emotions are associated with the left frontal area of the brain, whereas negative emotions are related to the right frontal area of the brain (Mohammadi et al., 2016). Mu Li et al. (Li & Lu, 2009) noticed that emotions could be classified with higher accuracy by using high-frequency waves, especially gamma waves. The work by D. Huang et al (Huang et al., 2012) illustrated two emotions (valence and arousal) by using beta and gamma bands.

In the last decade, researchers focused on the success of shallow machine learning and deep learning approaches to identify emotions using different features of EEG signals. Moreover, ensemble machine learning techniques with non-linear features remained unexplored. However, those approaches are less complex than deep learning and easy to apply. The sole purpose of this paper is to explore how emotions can be recognized using a smaller number of electrodes and ensemble method and improve accuracy. This paper proposes a non-linear features-based emotion detection method using eight ensemble methods, including SVM and KNN. We have evaluated six discrete emotions (hope, interest, excited, shamed, fear, and sad), where three emotions belong to positive emotions (hope, interest, and excited), and the other three are negative emotions (shamed, fear, and sad). Moreover, we consider the most popular feature selection method to select relevant electrodes associated with each emotion, namely the recursive feature elimination method. Fig. 1 depicts the framework of our suggested method, and the following are the contributions of this paper:

a) The study compared the performance of eight ensemble methods, including SVM and KNN, for each emotion and selected relevant hyperparameters for each classifier to boost the performance of the experiments.
b) We have utilized the synthetic minority oversampling (SMOTE) technique to handle the imbalanced data.
c) We have implemented recursive feature elimination (RFE) method to select the relevant number of electrodes for each emotion.
d) The experiments have been performed on two datasets to validate the model performance.

The remainder of this work is arranged in the following way. In the "related works" section, most of the recent and known research related to this work is discussed. Section 3.1 and section 3.2 presents the EEG dataset and data preprocessing, respectively. Target emotions and the data augmentation process are discussed in section 3.3. The "feature extraction" section describes the feature extraction process. In section 3.5, we have discussed the method for selecting the optimal electrodes related to each emotion. The classifiers and their working procedure are described in the "classification model" section. The "Experimental setup" section describes the process of choosing training and testing data and hyperparameters for each classifier. The performance of our proposed method and comparison of those results with other related works are discussed in the "Result and discussion" section. Moreover, statistical analysis is described in the Result and discussion section. In the conclusion section, the concluding remarks and future directions for future research are outlined.

2. Related work

Nowadays, numerous research fields such as education, healthcare, decision making, marketing, and more, are trying to build an emotional model for the development of the respective field because emotion recognition and understanding emotional response have direct implications on those fields. For example, emotion analysis in the education sector can assist in determining the attention level of the student and deciphering which part of the content the student finds difficult to understand. Moreover, it can help to increase the efficacy of the long-distance teaching process during the COVID-19 situation. Usually,

Table 1
Description of DEAP dataset.

Preprocessed signal using 32 electrodes	
Recorded Signals	Electroencephalogram (EEG)
Sampling Frequency	The signal was down sampled to 128 Hz and a bandpass frequency filter from 4.0 Hz to 45 Hz was applied.
Number of Channel	32 (AF3, AF4, C3, C4, Cz, CP1, CP2, CP5, CP6, F3, F4, F7, F8, Fz, FC1, FC2, FC5, FC6, Fp1, Fp2, O1, O2, Oz, P3, P4, P7, P8 Pz, PO3, PO4, T7 and T8)
No of stimulus	40 music videos
Artifacts	Common average referencing (CAR) and ocular artifacts (EOG) were removed by blind source separation algorithm.
Duration of signal	Total duration for each trail is 63 s where 60 s trial and 3 s pre-trial.
Number of samples of each channel	8064
Number of samples of each trial	1*32*8064 (video/trial*channel*data)
Number of samples per subject	40*32*8064 (video*channel*data)
Estimation of emotions using SAM mannequins in the range of 1–9 scales	Valence, arousal, dominance, and liking

EEG-based emotion recognition systems have involved in several steps such as signal acquisition, removing artifacts, extracting features, and designing a classification model. Several emerging approaches like machine learning and deep learning have been deployed to classify actual emotions in the past decade. In addition, researchers have been using different features from the time–frequency domain, time domain, frequency domain to understand the behavior of EEG with the change of emotions.

In the study (Taran & Bajaj, 2019), proposed multi-class least squares support vector machine was used (MC-LS-SVM) to identify four emotions (happy, fear, sad, and relax). They filtered the data using correlation-criterion and variational mode decomposition (VMD) methods and then they have used sample entropy, Tsallis entropy, Fractal dimension (higuchi), Hurst Exponent as input features to classify the emotions. Twenty-four channels were used to collect the EEG signals from 20 participants, whereas each of the participants watched movie clips as stimuli. The proposed method by the authors could achieve an overall accuracy of 90.63%. G. Chen et al. (Chen et al., 2020) have proposed SVM classifier for recognizing seven emotional states where five emotions (happiness, surprise, anger, neutral, sadness) were analyzed using TYUT 2.0 dataset and two (valence and arousal) emotions from the DEAP dataset. Six types of time and frequency-domain characteristics were calculated as features in order to track the EEG changes with the variation of emotions and the work could have achieved an average accuracy of 84.67% using DE features on the TYUT 2.0 dataset. In the study (Khare & Bajaj, 2021) proposed optimized variational mode decomposition was used for analyzing four emotions. Relevant electrodes were selected using the eigenvector centrality method (EVCM), and finally, one electrode was chosen. Four time-domain features (hjorth complexity, difference, absolute standard deviation value, log energy entropy, renyi entropy) were extracted as features to distinguish emotion, and each of the emotions has been classified using different classifiers.

Raja Majid Mehmood et al. have used the deep learning ensembles method for classification emotions (Mehmood et al., 2017). It is observed that they achieved an average accuracy of 76.7% by implementing Hjorth parameters. Similarly, J.X. Chen et al. (Chen et al., 2019) investigated emotions using EEG signals from the DEAP dataset. All the experiments are performed using some shallow machine learning methods (bagging tree, support vector machine, linear discriminant analysis, Bayesian linear discriminant analysis), and deep convolution neural network (CNN). They combined temporal and frequency features of EEG signals in both valence and arousal dimensions and achieved the highest accuracy using the CNN model. Yong Zhang et al. (Zhang et al., 2021) proposed a hierarchical multimodal emotion recognition framework to estimate emotional states in valence and arousal dimensions. This network performed based on CNN network on DEAP and MAHNOB-HCI databases. They achieved an average accuracy of 84.71% and 89% on the two corresponding datasets. In the work by Torres et al. (Torres et al., 2020), Random Forest (RF) and Deep Learning (DL) methods were applied for the classification of valence and arousal using eight electrodes and an average accuracy of 71.22% and 83.18%, respectively were obtained. In this article, mutual information matrix and chi-square statistics were implemented for reducing the dimensionality of the features, and entropy features were used as input to the classifier.

Oana Balan et al. (Bălan et al., 2020) applied various machine learning and deep learning techniques to classifying six basic emotions using the physiological recordings (EEG, EOG, EMG, GSR, and PPG) from the DEAP database. They also used fractal dimension and approximate entropy as a feature and three feature selection methods were applied to select the optimal features. Vijay et al. (Vijayan et al., 2015) extracted the auto-regressive coefficient from EEG signals to estimate different emotions such as happiness, fear, sadness, and hated. They applied the MCSVM classification algorithm and obtained an average classification accuracy of 94.097%. The proposed method has been applied to the publicly available DEAP dataset. Pane et al. (Pane et al., 2018) have implemented a decision tree algorithm to recognize four emotions (happy, angry, sad, and relax) using time and time–frequency domain features. They applied a bandpass IIR filter to remove artifacts from EEG signals. They separated different bands of frequencies of EEG signal and achieved an average accuracy of 81.64% using the DEAP database. It was also observed that relaxed emotion showed the highest accuracy considering five electrodes. The article (Gannouni et al., 2021) used the zero-time windowing-based epoch estimation (ZTWBES) algorithm to identify the epochs and relevant electrodes in each frequency band for each emotional state. Quadratic discriminant classifier (QDC) and RNN were applied to distinguish nine emotions using the DEAP database. The result showed an average accuracy obtained at 87.05%, 89.33%, and 86.53% using QDC, RNN-scheme 1, and RNN-scheme 2.

3. Methods and materials

In this article, we have developed a system for emotion analysis. Ten machine learning algorithms were employed to analyze the emotion using two datasets. The performance parameters like accuracy, precision, recall, F-measure, and AUC-ROC evaluate the performance of each classifier. All the experiments are performed using full features and an optimum subset of features. One feature selection method, namely, Recursive Feature Elimination (RFE), was used to choose the most relevant electrodes related to emotion. Then the SMOTE technique was used to balance the target attribute, and finally, the processed dataset was applied to each classifier. The parameters of each classifier was selected using the randomized grid search method. Moreover, cross-validation techniques were applied to validate the model performance.

Table 2
Description of AMIGOS dataset.

Preprocessed signal using 14 electrodes	
Recorded Signals	Electroencephalogram (EEG)
Sampling Frequency	The signal was down sampled to 128 Hz and bandpass frequency filter from 4.0 Hz to 45 Hz was applied.
Number of Channel	14 (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4)
No of stimulus	16 short music videos
Artifacts	Common average referencing (CAR)
Duration of signal	variable duration from 56 s to 155 s
Number of samples of each channel	Variable (7229–19886)
Number of samples of each trial	1*14*(7229–19886) (video/trial*channel*data)
Number of samples per subject	16*14*(7229–19886) (video*channel*data)
Estimation of emotions using self and external annotation	Valence, arousal, dominance, familiarity, liking (in the scale of 1 to 9) and seven basic emotions (binary value)

3.1. Dataset description

This paper conducted all the experiments for the proposed model using a publicly available DEAP dataset developed by researchers at Queen Mary University of London (Koelstra et al., 2012). Then another dataset named AMIGOS has been used to validate the model performance (Miranda-Correa et al., 2021). In this paper, we adopted the datasets mentioned above to assess the model's performance because both datasets evaluate emotions on the scale of 1 to 9 for valence and arousal. Moreover, both datasets collect EEG data using audio–video signals as stimuli to elicit emotions. The two publicly available datasets are described in the following section.

a) DEAP

It is a multimodal dataset that recorded EEG and other peripheral physiological signals of thirty-two subjects while each was watching 40 music videos which had a duration of 1 min for each video. They used a total of 48 electrodes during the recording of bioelectrical signals. Among 48 electrodes, 32 electrodes were used only for EEG recording. Each signal was recorded at a sampling frequency of 512 Hz. Participants evaluated each video based on levels of arousal, valence, like/dislike, dominance, and familiarity. The detail of the DEAP dataset for 32 channels is summarized in Table 1.

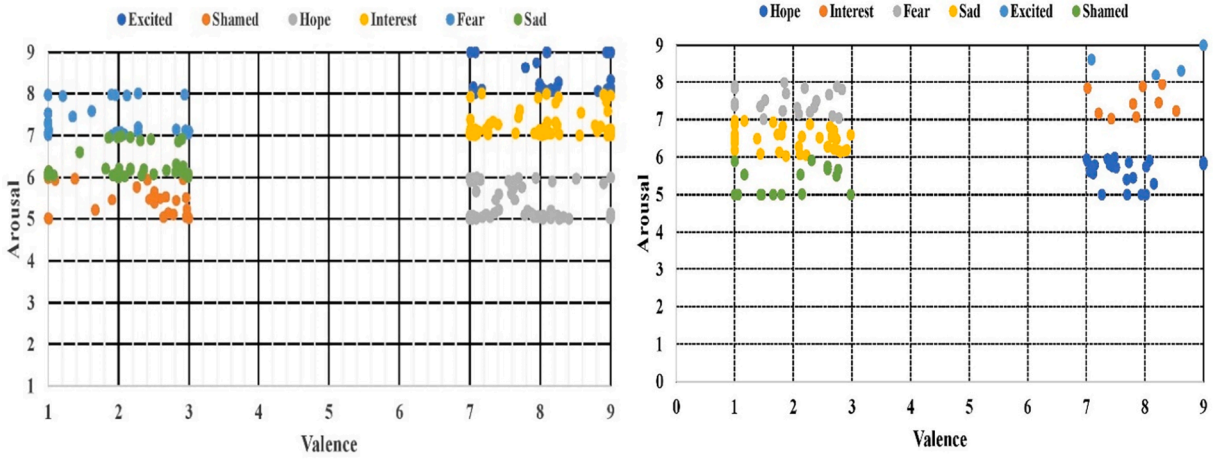
b) AMIGOS

This dataset provides two types of data using two experimental settings. In the first experiment, 40 participants watched 16 short videos with variable duration (<250 s), where all the participants watched those videos individually. In the second experiment, some participants (17) individually watched four long videos (duration >14 min), and some of them (20) watched the same videos in a group. Each group consisted of four members where the number of groups was five. In both situations, audio–video clips were used as stimulation materials to elicit the emotions, and three neurophysiological signals such as electroencephalogram (EEG), electrocardiogram (ECG), and galvanic skin response (GSR) were recorded during the stimulation of emotion. Total 17 channels were used to record the neurophysiological signals, 14 for EEG, another two for ECG (left and right), and rest one for GSR. The detail of the AMIGOS dataset for 14 channels is summarized in Table 2. The affective states of each of the subjects were evaluated using self (arousal, valence, dominance, liking, familiarity, and seven discrete emotions) and external annotation (valence and arousal).

3.2. Data preprocessing

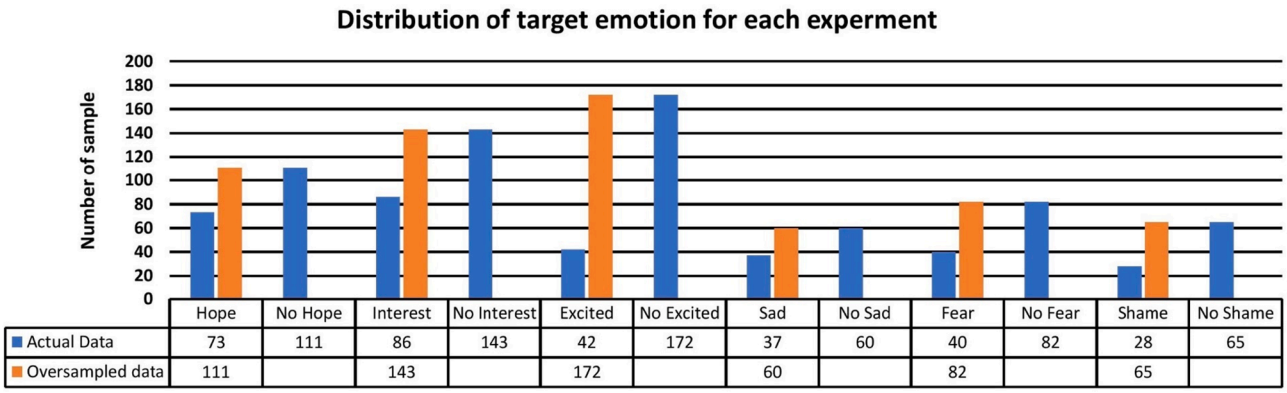
According to the DEAP dataset, the total number of samples is 8064, which means each signal's duration is 63 s. This study used the last 60 s of the signals by removing a 3 s pre-trial baseline. Therefore, the total number of samples of each trial per channel is 7680. Then, encouraged by the following research (Zhang et al., 2016; Mohammadi et al., 2016), we divided 7680 samples into 20 segments. The total number of samples of each segment was 384, and the duration of each segment was 3 s. Therefore, each signal of the channels was segmented into 20 signals with a duration of 3 s, and the total number of samples of each subject was 9,830,400 (20(segment) *384(samples/epoch) *32(channels) *40 (trial)). Similarly, each experiment's total number of samples was 9830400(samples/subject) *32(subject).

Similarly, we used preprocessed data from the AMIGOS dataset, with



(a) DEAP (b) AMIGOS

Fig. 2. Distribution of each emotion based on valence and arousal dimensions using the (a) DEAP and (b) AMIGOS dataset. Each of the colors represents a different emotion.



	Hope	No Hope	Interest	No Interest	Excited	No Excited	Sad	No Sad	Fear	No Fear	Shame	No Shame
Actual Data	73	111	86	143	42	172	37	60	40	82	28	65
Oversampled data	111		143		172		60		82		65	

Fig. 3. Distribution of each emotion using SMOTE technique on DEAP dataset.

different signal durations. It contains 40 files, and each file represents an individual participant's EEG signal using 14 channels. Among 40 participants, the information of six participants (12, 21, 22, 23, 24, and 33) were inconsistent. For example, the signal for trial 5 of participant 12 was missing. Therefore, the data obtained from participants 23 and 33 are completely ignored. However, Data obtained from subjects 12, 21, 22, 24 are partially attached to the valid data. Since the duration of each signal is different, we removed the first 5 s pre-trial baseline and selected the possible shortest duration of the signal. Therefore, we have chosen 52 s of signals to unify the standard for the experiment. Hence, the total number of samples of each segment was 384, and the duration of each segment was 3 s. Therefore, each signal of the channels was segmented into 14 signals with a duration of 3 s, and the total number of samples of each subject was 1,204,224 (14(segment) *384(samples/epoch) *14 (channels) *16(trial)). Similarly, each experiment's total number of samples was 1,204,224 (samples/subject) *40(subject). After completing the process of signal segmentation, we normalized each segmented signal of each channel using the max–min normalization technique by equation (1). The purpose of normalization is to make the same scale (0 to 1) for both datasets because each feature is equally essential.

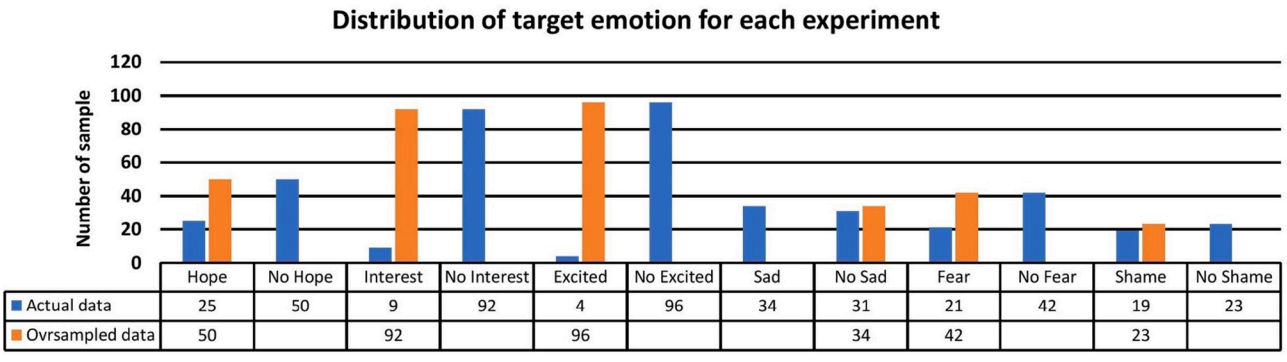
$$EEG_{scaled} = \frac{EEG - \min(EEG)}{\max(EEG) - \min(EEG)} \left(new_{\max(EEG)} - new_{\min(EEG)}\right) + new_{\min(EEG)} \tag{1}$$

3.3. Target emotions and data augmentation

Emotions come in various forms and have an impact on how we live and connect with others. Psychologists have also attempted to categorize people's various sorts of emotions. To describe and explain emotions, a few distinct ideas have been evolved. In the past few decades, psychologists introduced emotions in two ways: the discrete emotional model and the dimensional model (two dimensional and three dimensional). On the other hand, researchers attempted to assess emotions using two ways: subjective measure (self-assessment manikin and questionnaires) and physiological theories of emotion (physiological signals) (Qing et al., 2019). However, these discrete and categorical emotions usually are estimated using the valence, arousal, and dominance model (VAD). Moreover, several researchers showed that emotions might be classified into three groups: positive, negative, and calm, which depend on valence and arousal dimensions. Beatriz García-Martínez et al. (García-Martínez et al., 2016) explained calm and negative stress from EEG recordings using entropy-based metrics. Furthermore, Chunmei Qing et al. (Qing et al., 2019) added positive emotion with negative and calm states in their research. All three emotional states are evaluated in the valence and arousal dimensions.

In this paper, six discrete emotions (hope, interest, excited, sad, fear, and shamed) that belong to positive and negative emotions have been evaluated in the valence and arousal dimension using the DEAP and AMIGOS dataset. In both datasets, each of the subjects rated each video in valence, arousal, and dominance on a scale of 1 to 9. However, according to Mehrabian and Russell's theory, the rating values were −1 to 1 for valence, arousal, and dominance. Therefore, we converted the scale of both dataset from −1 to 1 to obtain the same scale (Bălan et al., 2019). We have considered the two methodologies that were described in the article (Mehrabian & Russell, 1977) and (Qing et al., 2019) to estimate six emotions. According to the methods mentioned above, the distribution of each emotion in valence and arousal dimensions using the DEAP and AMIGOS dataset is depicted in Fig. 2.

It has also been observed that the number of examples in the training dataset for each class label is imbalanced. For example, the number of samples in the majority class of the DEAP dataset (sad-60) is much higher than that of the minority class (sad-37). Imbalanced classifications offer difficulty for predictive modeling, and that results in poor predictive performance in the models. In this work, one approach known



	Hope	No Hope	Interest	No Interest	Excited	No Excited	Sad	No Sad	Fear	No Fear	Shame	No Shame
Actual data	25	50	9	92	4	96	34	31	21	42	19	23
Ovrsampled data	50		92		96			34	42		23	

Fig. 4. Distribution of each emotion using SMOTE technique on AMIGOS dataset.

as the synthetic minority oversampling technique (SMOTE) proposed by Chawla et al. (Chawla et al., 2002) has been used to overcome this difficulty. Generally, SMOTE is an oversampling technique in which synthetic samples for the minority class are generated. This method increases the amount of data by duplicating samples in the minority class and does not give any new data to the machine learning model. Similarly, this approach assists in overcoming the problem of overfitting caused by random oversampling. The following statements outlined how synthetic samples are created. Firstly, the sample and its nearest neighbor is considered. Then the distance metric method is used to make a difference between the sample and its nearest neighbor. Finally, this difference is multiplied by any random value in the range [0,1] and added to the previously computed feature vector. Fig. 3 and Fig. 4 shows that the minority classes of emotion have been oversampled to balance the data distribution using SMOTE technique.

3.4. Feature extraction

Several feature extraction methods such as time, time–frequency, frequency, and non-linear analysis have been used to understand and represent the non-linear behavior of EEG signals in the last few decades. Furthermore, non-linear features received more attention from the affective computing community in identifying emotions because it helps us explain and represent the EEG signals more accurately than the other methods. Xiao Wei Wang et al. (Wang et al., 2014) used time-domain features as well as approximate entropy and fractal dimension. They obtained an average accuracy of 85% using 62 electrodes while six participants watched video clips to assess two emotional states (positive and negative). Jie X et al. (Jie et al., 2014) extracted sample entropy to classify two emotions of 32 subjects using 32 electrodes. Similarly, Z. Lan et al. (Lan et al., 2015) distinguished four basic emotions of 14 participants using the statistical features and fractal dimension. In this paper, we have implemented three non-linear approaches, namely Higuchi fractal dimension, sample entropy, and permutation entropy, to represent the distinctive characteristics of EEG signals.

a) Higuchi fractal dimension.

The Higuchi fractal dimension indicates the degree of irregularity and complexity of time-series signals. Higuchi first introduced this approach in 1988 (Higuchi, 1988). Furthermore, this approach has been widely used as a complexity measure of signals in various fields, including speech, medicine, and biomedical applications. Moreover, numerous studies have been successfully conducted to detect emotions using the Higuchi fractal dimension in the last few years. The complexity of the EEG signal of N sample data can be measured using the Higuchi fractal dimension of the following process:

Firstly, a new time series x_k^m is constructed from a finite number of EEG samples (N = 7680 for each channel) as follows:

$$x_k^m : x(m), x(m+k), x(m+2k), x(m+3k), \cdots\cdots\cdots, x\left(m+\left[\frac{N-m}{k}\right].k\right) \tag{2}$$

where m = 1, 2, 3, 4… k, and m and k denote the beginning and interval times, respectively, and [ ] signifies the Gauss' notation.

The length for each of the k time series x_k^m is as follows.

$$L_m(k) = \frac{1}{k}\left\{\left(\sum_{i}^{\frac{N-m}{k}} |x(m+ik) - x(m+(i-1).k)|\right)\frac{N-1}{\left[\frac{N-1}{k}\right].k}\right\} \tag{3}$$

Then the average length of the curve for each time interval k can be calculated as follows.

$$L(k) = \frac{1}{k}\sum_{m=1}^{k} L_m(k) \tag{4}$$

The term $\frac{N-1}{\left[\frac{N-1}{k}\right].k}$ represents the normalization factor and if $L(k) \propto k^{-D}$, the curve is therefore fractal with the dimension D.

b) Sample entropy.

Sample entropy is nothing but a negative algorithm of conditional probability in which two sequences similar for two points remain identical at the following (m + 1) points, and self-matches are excluded from the probability calculation (Richman & Moorman, 2000). Moreover, Sample entropy shows relative consistency and is largely unaffected by the series' length. Traditionally, sample entropy estimates the randomness or irregularity of the time series signal. This method has been widely used in different fields of research, especially for physiological signals (García-Martínez et al., 2016), (X et al., 2014), (Gao et al., 2019; Huang et al., 2013; Jiang et al., 2015). The sample entropy can be defined by the following equation as follows.

$$SampEn(m, r, N) = -\ln\left[\frac{B_{m+1}(r)}{B_m(r)}\right] \tag{5}$$

Where B_m can be calculated as follows.

$$B_m(r) = \frac{\left\{number\ \ of\ \ all\ \ proable\ \ pairs(i,j)\ \ with\ \ \left|x_i^m - x_j^m\right| \left\langle r, i \neq j\right\}\right.}{\{(N-m+1)(N-m)\}} \tag{6}$$

where $\left(x_i^m - x_j^m\right)$ denotes the distance between x_i^m and x_j^m, N is the length of the data, r is the tolerable standard deviation of sequence of data and m is the embedded dimension. It is found that the recommended range for r is usually between 0.1 and 0.25σ and the value of m should be in the range between 1and 2.

c) Permutation entropy.

Bandt and Pompe (Bandt & Pompe, 2002) proposed permutation entropy (PE) in 2002, and PE is known for being computationally efficient and useful for real-time applications compared to other entropy methods. In addition, the irregularity of any signal, including regular, non-linear, non-stationary, and chaotic can be calculated in this method. Numerous researches have been used permutation entropy in physiological signal analysis in various fields of biomedical engineering. On the other hand, emotion detection through the analysis of permutation entropy remains unexplored. Therefore, we have evaluated permutation based on Shannon entropy to estimate emotions in this paper, and it can be expressed by the following equation.

$$PE(x, m, N, \tau) = -\sum_{j=0}^{m!-1} p_j log_2 p_j \tag{7}$$

where, x is a time series, m is embedding dimension, τ is the delay time, and N is the number of samples for a given time series. We can evaluate the permutation entropy of a definite time series $\{x_i\}_{i=1,2,3,\cdots,N}$ of length N using the following steps (Riedl et al., 2013):

i. Consider the embedding dimension m for permutation or ordinal pattern π_j^m, where the $\pi_j^m = \pi_0, \pi_1, \pi_2 \cdots \pi_{m!-1} (j = 0, 1, 2, \cdots, (m! - 1))$
ii. Set $i = 1$ as the starting point of the selected time series $\{x_i\}_{i=1,2,3,\cdots,N}$ and $c_j = 0$ for each π_j, where, c_j is the counter of permutation pattern.
iii. Calculate the subsequence $x_{i,(i+1),\cdots\cdots(i+n-1)}$ according to m and also find the ranks corresponding to the values in the sequence. The ranks, as well as subsequence, are sorted in ascending order. The rank values are associated with the subsequence values. For example, subsequence values = 2,4,3 (rank value = 0,2,1), after ascending order subsequence values will be 2,3,4, therefore, new rank values will be 0,1,2.

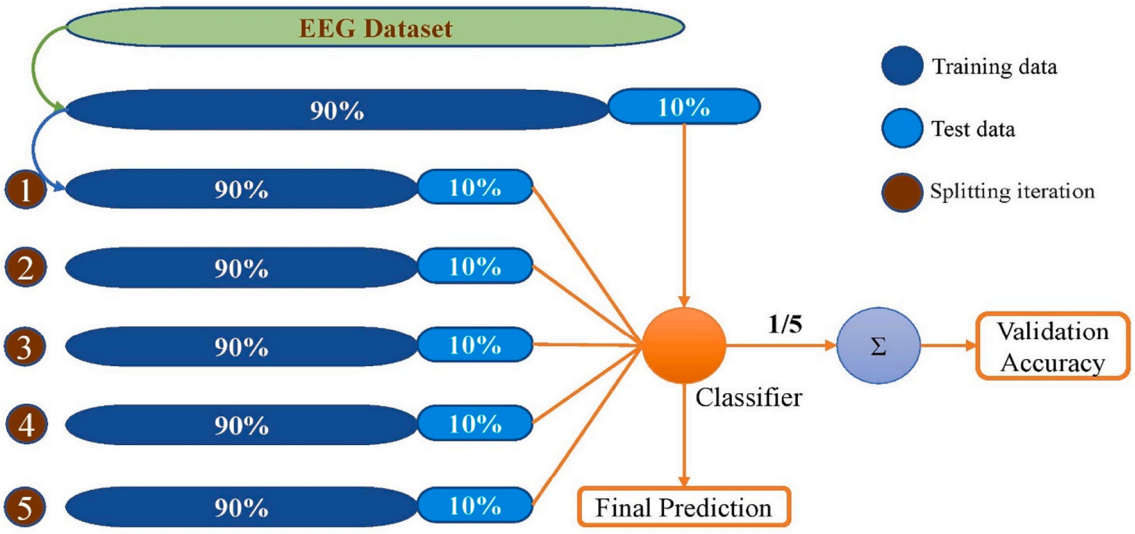


Fig. 5. Shuffle-split cross-validation technique.

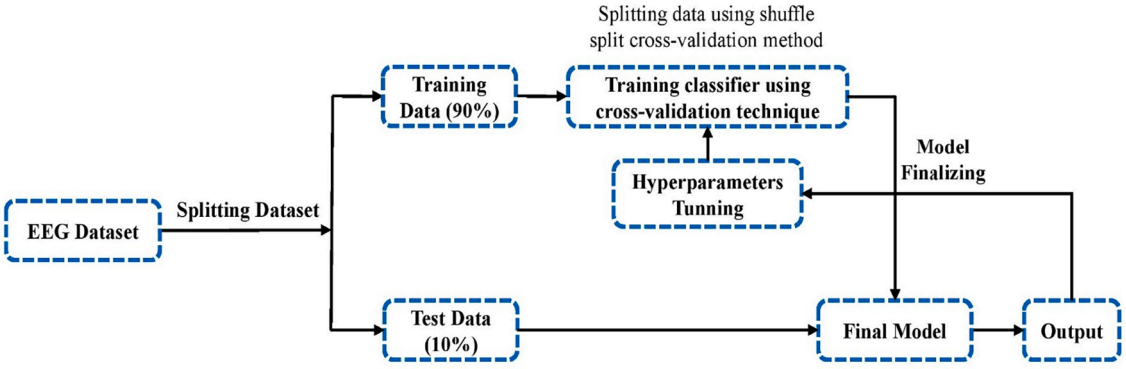


Fig. 6. Process of hyperparameter tunning using the randomized grid search cross-validation method.

iv. Find the similarity of rank sequence $(r_i r_{(i+1),\ldots\ldots,} r_{(i+n-1)})$ from step 3 to the all-possible permutation patterns $\pi_j^m = \pi_0, \pi_1, \pi_2 \cdots .. \pi_{m!-1}$. if similarity occurs, then increase the counter for equal pattern $\pi_a = r_i r_{(i+1),\ldots\ldots,} r_{(i+n-1)}$ by $c_a = c_a + 1$.
v. If $i < (N-m+1)$ then increase i by $i = i+1$ and start step 3 again. If $i \geq (N-m+1)$ then go to the next step.
vi. Evaluate the probability of each permutation pattern by the following equation $p_j = \frac{no\ of\ equal\ pattern\ for\ each\ \pi_j^m(c_j)}{Total\ number\ of\ equal\ patterns\ (\sum c_a)}$
vii. Finally, calculate the permutation entropy using equation (7).

3.5. Selection of optimal electrodes to each emotion

Numerous strategies, namely, filtering, wrapping, embedded, and hybrid method, have been discussed in the analysis of EEG-based emotion analysis for the selection of channels. Indeed, electrodes reduction can improve learning performance, reduce computational complexity, provide more generalizable models (Alotaiby et al., 2015). Moreover, it not only reduces the dimensionality of the dataset but also helps the algorithm learn faster. Some researchers have tried to find the relation between the brain region and emotions (Jenke et al., 2014). In this paper, we have focused only on those electrodes which are relevant to each emotion. To fulfill this requirement, we have implemented one of the popular feature selection techniques called recursive feature elimination, to make a new subset of electrodes. In this method, RFE ranks the features based on their importance and returns top N features after eliminating the least important features (Q. Chen et al., 2018). The top features relevant to the class label can be calculated based on their importance from training data using the following steps:

i. Set the training data $\{x_1, x_2, \cdots\cdots .., x_n\}$ with known class labels $\{y_1, y_2\}$ to the supervised model.
ii. Train the model.
iii. Compute the importance of all the features.
iv. Remove the least important features from the list of features using the feature ranking criterion.
v. Rebuild the model with remaining features.
vi. If feature subset > 0, go to step 3 and repeat the process until it reaches zero features

3.6. Classification model

We have used different types of ensemble machine learning algorithms for each type of EEG feature to classify emotional states. Moreover, we have adopted two shallow machine learning techniques: support vector machine, k-nearest neighbor. The following are the steps involved in building a classification model:

i) Set up the classifier that will be used
ii) Train the classifier for hyper parameter tunning
iii) Rebuild the model and train again using best parameter
iv) Predict the emotional states

Table 3
Hyperparameters of an individual classifier for binary classification.

Algorithm	Hyperparameter	Range and Name
Support vector machine	**C**: penalty parameter for regularization	0.1, 1, 10, 100
	Gamma: kernel Co-efficient	0.01, 0.1, 1, 10, 100
	Kernel: radial basis function (rbf)	–
K-nearest neighbor	**n_neighbors**: number of neighbors	1–10 (odd value)
Random Forest	n_estimators: number of trees	50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 400
	max_features: number of features for the best split.	auto
	bootstrap	True, False
	min_samples_split: minimum number of samples to split an internal node.	2–8
	min_samples_leaf: minimum number of samples required to be at a leaf node	1–5
	max_depth: maximum depth of the tree	None
Voting	base_estimator: list of classifiers	Support vector machine and k-nearest neighbor
	voting	soft
Bagging	base_estimator: list of classifiers	Decision tree
	n_estimators: number of base estimators	5, 10, 15, 20
Adaptive boosting	base_estimator: classifier	Decision tree
	n_estimators: maximum number of estimators	50, 60,70, 80, 100
	learning rate: how slowly or fast classifier update its last value.	0.0001, 0.001, 0.01, 0.1, 1
Gradient boosting	n_estimators: number of boosting states	100–500
	max_depth: maximum depth of the tree of each classifier	2–8
	min_samples_leaf: minimum number of samples required to split an internal node	4, 8, 10, 16, 20
	learning rate: how slowly or fast classifier update its last value.	0.01, 0.05, 0.1
Extreme gradient boosting	**n_estimators**: maximum number of estimators	100–700
	max_depth: maximum depth of the tree	2–10
	min_child_weight	1–9
	learning_rate: how slowly or fast classifier update its last value.	0.05, 0.1, 0.15, 0.2, 0.25
LightGBM	**n_estimators**: maximum number of estimators	100–500
	max_depth: maximum depth of the tree	2–10
	min_child_weight	1–9 (odd value)
	learning rate: how slowly or fast classifier update its last value.	0.05, 0.1
Stacking	estimators: Base estimators which will be stacked together	Support vector machine and k-nearest neighbor
	final_estimator: A classifier which will be used to combine the base estimators.	Quadratic discriminant analysis

v) Evaluate the classifier performance for each of the emotional states

a) Support vector machine

Support vector machine is such a type of an algorithm where two classes are identified by finding the hyperplane. The hyperplane separates the two classes, and it is used in such a way that it can maximize the distance between two classes. Many researchers prefer the support vector machine because it produces excellent accuracy using less computing power. In this paper, we have performed binary classification using non-linear SVC with radial basis function (RBF) kernel (Schölkopf, 1998). The RBF kernel can be expressed by the following equation.

$$k(x,y) = e^{-\left(\frac{\|x-y\|^2}{2\sigma^2}\right)} \tag{8}$$

b) K-nearest neighbor

K-nearest neighbor, a non-parametric and lazy learner, is widely used in classification due to its simplicity and ease of implementation. KNN classifies the class levels by identifying the similarity between the data points (training and test data) This algorithm can measure similarity using distance measurement (Euclidean, Manhattan, Minkowski, Mahalanobis) or cosine similarity between training and test data points. In this paper, we calculated the Euclidean distance (Altman, 1992) between training (x_i^a and test y_i^b) data points to find the closest neighbors, and it can be expressed as follows:

$$d\left(x_i^a, y_i^b\right) = \sqrt{\sum_{i=1}^{n} \left(x_i^a - y_i^b\right)^2} \tag{9}$$

The KNN algorithm supports the usage of a majority vote to do the classification, and the value of the nearest neighbor (k) usually is an odd number to avoid ambiguity (Kotsiantis et al., 2007).

c) Random forest

Random forest algorithm creates decision trees on various subsets of given data samples and then gets the prediction from each of them and finally selects the best solution using voting. It is an ensemble method developed by Breiman (Breiman, 2001) that is superior to a single decision tree because it reduces the overfitting problem by averaging the results of different decision trees. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

d) Weighted majority voting

Ensemble learning is the process of combining predictions from various classifiers. The most basic and widely used combination method is majority voting. In this method, each classifier can be trained using different splits of the same training dataset and same algorithm or using the same dataset with different algorithms or any other method. Then it considers the majority number of votes from the classifier and shows the prediction by updating the weights of classifiers (Dogan, n.d.). In the case of majority voting, the final prediction for n number of classifiers can be explained with the following equation.

$$\widehat{y} = \max_{1\le j\le k} \sum_{i=1}^{n} w_i p_{i,j} \tag{10}$$

Where, $p_{i,j}$ is the predicted class of the i^{th} classifiers for class label j, w_i is the weighting parameter for i^{th} classifiers and $\widehat{y}$ is the final prediction. The prediction of each classifier is denoted by either 0 or 1 ($p_{i,j} \in \{0,1\}$). Voting classifiers often use one of two voting systems: hard voting and soft voting. In hard voting, each classifier votes for a class, and the majority wins, whereas, in soft voting, each classifier assigns a probability value and the result is the highest average of all the class probabilities. In this paper, the support vector machine (SVM) and k-nearest neighbor (KNN) were the categorization methods employed to analyze the majority voting ensemble (See Figure).

e) Bagging

In the year 1994, Leo Breiman (Breiman, 1996) proposed this algorithm, and it was known as bagging predictors. Bagging is the process of fitting multiple decision trees on different training data samples where

Table 4
Validation (mean ± std. dev.) and test accuracy (%) using DEAP dataset without RFE.

Algorithm	Higuchi Fractal Dimension												
	Sad		Fear		Shamed		Hope		Interest		Excited		Average (test)
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	
SVM	91.67 ± 2.23	92.5	89.93 ± 1.23	92.68	93.93 ± 1.09	94.61	83.55 ± 1.44	84.68	86.29 ± 0.89	84.26	94.65 ± 0.47	95.05	90.63
KNN	91.02 ± 2.20	91.66	88.92 ± 0.69	87.80	90.94 ± 1.36	93.84	81.40 ± 1.27	84.90	84.58 ± 1.55	84.09	93.74 ± 0.73	93.16	89.24
RF	90.74 ± 1.87	90.41	88.78 ± 1.34	89.32	92.05 ± 1.10	92.69	81.80 ± 0.73	83.55	84.54 ± 0.72	85.13	94.77 ± 0.63	95.78	89.48
Voting	92.96 ± 2.10	93.33	89.80 ± 0.54	90.54	94.10 ± 1.39	96.15	84.40 ± 0.99	86.71	87.07 ± 1.59	85.31	95.58 ± 0.46	95.78	91.30
Bagging	88.98 ± 2.59	88.75	88.18 ± 0.57	88.41	91.28 ± 2.24	91.92	80.85 ± 0.85	79.95	82.87 ± 0.83	82.69	93.13 ± 0.58	92.44	87.36
AdaBoost	92.41 ± 1.33	89.58	89.66 ± 0.79	90.24	94.10 ± 1.56	95.38	82.45 ± 0.93	84.23	85.40 ± 1.76	84.09	95.06 ± 0.86	95.63	89.86
GBDT	91.20 ± 1.37	88.75	88.11 ± 1.01	89.32	92.48 ± 2.12	94.23	81.75 ± 1.21	81.98	84.39 ± 2.09	83.21	93.58 ± 0.75	94.62	88.69
XGBoost	91.20 ± 1.28	89.16	87.09 ± 0.75	88.10	91.45 ± 1.27	93.84	79.30 ± 1.35	80.63	83.11 ± 1.89	81.64	93.55 ± 0.88	93.89	87.88
LightGBM	89.72 ± 2.58	88.75	87.91 ± 0.39	88.41	90.94 ± 1.13	92.69	78.85 ± 1.52	80.63	82.37 ± 0.81	81.81	94.00 ± 0.93	94.76	87.84
Stacking	91.85 ± 2.08	92.91	89.86 ± 1.25	92.68	95.30 ± 1.35	96.92	83.75 ± 1.44	84.90	87.53 ± 1.05	85.13	96.68 ± 0.44	96.65	91.53
Algorithm	**Sample Entropy**												
SVM	88.15 ± 1.67	91.25	90.27 ± 2.00	90.85	91.71 ± 1.54	90.38	80.15 ± 1.79	79.50	83.03 ± 1.91	82.69	96.97 ± 0.93	96.22	88.48
KNN	86.57 ± 1.66	84.58	83.51 ± 1.03	82.92	88.63 ± 0.96	90.00	76.05 ± 1.18	78.82	80.31 ± 1.71	80.06	89.90 ± 0.81	89.82	84.37
RF	86.85 ± 2.51	87.08	85.61 ± 1.29	84.14	90.26 ± 3.07	89.23	77.60 ± 1.21	75.67	81.94 ± 1.22	82.34	93.81 ± 1.42	91.13	84.93
Voting	88.33 ± 2.02	90.83	88.58 ± 0.92	88.71	92.99 ± 1.96	93.84	79.65 ± 0.60	80.85	84.04 ± 1.84	82.16	95.13 ± 1.11	94.18	88.43
Bagging	84.63 ± 2.63	84.16	83.58 ± 1.52	83.84	87.78 ± 3.64	88.46	78.75 ± 1.14	76.57	79.65 ± 2.25	80.59	92.71 ± 1.63	90.26	83.98
AdaBoost	88.15 ± 3.05	90.00	87.70 ± 0.92	85.97	91.28 ± 2.43	92.69	69.50 ± 0.82	65.54	83.65 ± 2.36	83.91	94.77 ± 1.41	93.31	85.24
GBDT	85.93 ± 2.63	87.91	86.01 ± 1.59	84.45	89.32 ± 2.13	90.76	77.75 ± 1.17	74.45	81.32 ± 1.99	81.29	95.52 ± 1.24	91.86	85.12
XGBoost	85.28 ± 1.97	87.91	85.61 ± 1.65	85.67	88.72 ± 2.79	88.46	77.50 ± 1.15	74.32	80.39 ± 1.83	79.02	92.32 ± 1.18	90.55	84.32
LightGBM	83.89 ± 3.27	83.75	83.92 ± 1.62	85.06	88.12 ± 3.61	88.46	76.35 ± 1.49	76.12	81.55 ± 1.45	80.76	91.94 ± 1.78	90.98	84.19
Stacking	88.89 ± 2.21	92.08	89.73 ± 1.83	90.24	93.59 ± 1.64	94.61	80.80 ± 1.53	80.40	84.16 ± 2.10	82.69	96.90 ± 0.90	96.36	89.40
Algorithm	**Permutation Entropy**												
SVM	85.93 + 1.67	82.50	89.39 + 1.85	90.85	88.97 + 1.70	88.84	79.25 + 0.97	80.18	81.32 + 0.95	83.39	93.48 + 0.75	92.73	85.15
KNN	88.80 + 1.39	83.75	85.74 + 2.36	83.53	88.12 + 2.17	85.38	78.65 + 1.74	81.75	81.09 + 1.51	80.76	90.00 + 1.50	90.55	83.03
RF	89.81 + 3.07	86.66	87.91 + 2.58	88.10	88.63 + 2.72	87.69	80.25 + 1.40	81.08	81.17 + 2.14	82.51	94.68 + 0.95	95.49	85.21
Voting	91.39 + 1.23	88.75	89.46 + 1.31	88.71	90.51 + 2.84	90.76	81.10 + 1.93	82.88	83.42 + 2.15	83.56	94.03 + 0.82	94.04	86.93
Bagging	87.59 + 3.03	85.83	86.96 + 1.61	85.97	87.95 + 3.39	86.53	79.40 + 0.98	81.30	80.43 + 1.99	82.69	92.35 + 0.83	93.31	84.46
AdaBoost	91.11 + 1.93	89.58	89.73 + 0.70	89.63	91.45 + 2.21	90.38	81.05 + 2.12	82.43	82.10 + 2.46	84.79	94.35 + 1.10	96.07	87.36
GBDT	86.67 + 1.72	85.83	87.03 + 1.83	89.32	88.97 + 1.61	89.61	79.20 + 1.66	80.63	79.73 + 1.29	79.54	93.10 + 1.15	94.76	84.99
XGBoost	87.22 + 2.76	87.08	86.55 + 1.88	85.67	88.38 + 2.84	87.30	78.20 + 2.06	79.72	80.70 + 1.57	80.41	91.94 + 0.93	94.04	84.04
LightGBM	87.59 + 2.06	86.25	86.69 + 0.97	86.89	89.23 + 3.61	85.76	77.45 + 2.36	77.92	79.88 + 1.82	79.19	92.68 + 1.21	94.47	83.20
Stacking	90.74 + 1.49	87.08	89.73 + 1.80	92.37	89.91 + 2.67	89.61	81.10 + 2.32	83.10	81.13 + 2.08	84.44	94.74 + 0.42	95.93	87.32

different training subsets are taken randomly with replacement. To construct data subsets for training the individual base classifier, it uses bootstrap sampling. Some data points may be repeated from the original training data points in each bootstrap sample. Once the algorithm is trained on all the bootstrap samples, the most voted class is accepted. Assume that n is the number of bootstrap samples of size k, where k is the total number of training samples. Each bootstrap is used to train the base classifier c_t where $t = 1, 2, 3, \cdots, m$. Therefore, the probability of not choosing ($P_{duplicate}$) the data point in the particular position of the bootstrap sample from the training dataset may be described with equation (11).

$$P_{duplicate} = \left(1 - \frac{1}{k}\right)^k \tag{11}$$

Table 5
Precision, recall, F-measure and AUC-ROC (%) using DEAP dataset without RFE.

Algorithm	Higuchi Fractal Dimension																							
	Sad				Fear				Shamed				Hope				Interest				Excited			
	P	R	F	AR	P	R	F	AR	P	R	F1	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR
SVM	96.36	86.17	90.98	91.38	91.35	93.08	92.21	92.39	96.46	90.83	93.56	93.98	91.70	77.97	84.28	85.30	88.88	81.97	85.29	85.97	97.23	97.83	97.53	97.68
KNN	87.96	95.12	91.40	90.72	82.10	98.11	89.39	88.99	93.33	93.33	93.33	93.80	81.00	92.07	86.18	84.74	79.63	92.57	85.62	84.69	87.77	97.53	92.39	92.72
RF	91.59	88.61	90.08	90.03	86.82	91.19	88.95	89.08	92.37	90.83	91.59	92.22	86.03	84.14	85.07	84.92	81.53	82.68	82.10	82.17	95.10	95.98	95.54	95.79
Voting	91.20	92.68	91.93	91.64	88.43	96.22	92.16	92.19	93.38	94.16	93.77	94.22	86.69	88.98	87.82	87.35	83.77	89.39	86.49	86.22	95.23	98.76	96.96	97.18
Bagging	87.80	87.80	87.80	87.49	85.63	93.71	89.48	89.45	91.52	90.00	90.75	91.42	83.47	84.58	84.02	83.53	78.36	84.45	81.29	80.80	91.44	95.67	93.51	93.85
AdaBoost	91.80	91.05	91.42	91.25	89.34	94.96	92.07	92.15	93.22	91.66	92.43	92.97	86.89	97.66	87.28	86.92	80.32	87.98	83.97	83.43	94.01	96.91	95.44	95.76
GBDT	89.07	86.17	87.60	87.53	87.35	95.59	91.29	91.28	92.37	90.83	91.59	92.22	86.16	85.02	85.58	85.36	79.93	83.03	81.45	81.31	93.73	96.91	95.29	95.57
XGBoost	88.33	86.17	87.24	87.10	85.71	90.56	88.07	88.18	90.90	91.66	91.28	91.90	83.04	84.14	83.58	83.08	78.83	81.62	80.20	80.08	91.81	96.91	94.29	94.61
LightGBM	87.50	85.36	86.41	86.27	86.54	93.08	89.69	89.73	90.16	91.66	90.90	91.54	83.84	84.58	84.21	83.76	77.70	81.27	79.44	79.21	92.35	96.91	94.57	94.88
Stacking	93.27	90.24	91.73	91.70	91.30	92.45	91.87	92.08	94.87	92.50	93.67	94.10	91.87	79.73	85.37	86.18	88.88	81.97	85.29	85.97	98.13	97.53	97.83	97.94
Algorithm	**Sample Entropy**																							
SVM	98.05	82.11	89.38	90.20	90.19	86.79	88.46	88.95	86.36	95.00	90.47	91.07	89.50	71.36	79.41	81.30	88.04	78.09	82.77	83.85	97.81	96.91	97.36	97.49
KNN	86.77	85.36	86.06	85.84	79.18	98.11	87.64	86.92	85.82	95.83	90.55	91.13	80.38	74.00	77.06	77.55	75.14	93.99	83.51	81.77	82.63	96.91	89.20	89.39
RF	89.65	84.55	87.02	87.14	85.53	85.53	85.53	85.96	90.75	90.00	90.37	91.07	80.47	74.44	77.34	77.77	81.22	84.09	82.63	82.53	91.94	95.06	93.47	93.82
Voting	92.62	91.86	92.24	92.08	83.14	93.08	87.83	87.66	92.12	97.50	94.73	95.17	82.66	81.93	82.30	81.98	80.12	91.16	85.28	84.51	92.92	97.22	95.02	95.31
Bagging	84.00	85.36	84.67	84.13	80.48	83.01	81.73	82.04	82.44	90.00	86.05	86.78	78.53	75.77	77.13	77.05	77.04	86.57	81.53	80.65	88.57	95.67	91.98	92.34
AdaBoost	90.75	87.80	89.25	89.20	85.00	85.53	85.26	85.66	90.16	91.66	90.90	91.54	79.91	78.85	79.37	79.05	79.68	88.69	83.94	83.27	92.05	96.60	94.27	94.59
GBDT	88.33	86.17	87.24	87.10	82.20	84.27	83.22	83.55	85.71	90.00	87.80	88.57	76.21	76.21	76.21	75.66	78.89	85.86	82.23	81.68	88.88	96.29	92.44	92.79
XGBoost	91.81	82.11	86.69	87.21	82.82	84.90	83.85	84.16	84.80	88.33	86.53	87.38	79.35	76.21	77.75	77.73	78.13	85.86	81.81	81.16	87.25	97.22	91.97	92.29
LightGBM	88.79	83.73	86.19	86.31	80.83	84.90	82.82	82.98	83.33	87.50	85.36	86.25	77.06	74.00	75.50	75.48	76.77	84.09	80.26	79.59	88.41	96.60	92.33	92.67
Stacking	94.54	84.55	89.27	89.71	90.25	87.42	88.81	89.27	93.38	94.16	93.77	94.22	87.50	70.92	78.34	80.16	87.26	82.33	84.72	85.28	98.11	96.60	97.35	97.47
Algorithm	**Permutation Entropy**																							
SVM	88.97	91.86	90.40	89.95	86.28	94.96	90.41	90.38	85.29	96.66	90.62	91.19	79.50	85.46	82.37	81.21	81.27	85.86	83.50	83.24	99.35	95.67	97.48	97.56
KNN	88.70	89.43	89.06	88.73	75.84	98.74	85.79	84.57	82.83	92.50	87.40	88.03	75.09	88.98	81.45	79.05	78.45	82.33	80.34	80.09	84.25	99.07	91.06	91.29
RF	88.33	86.17	87.24	87.10	86.66	89.93	88.27	88.45	89.83	88.33	89.07	89.88	83.01	77.53	80.18	80.47	80.91	80.91	80.91	81.11	94.47	95.06	94.76	95.05
Voting	86.15	91.05	88.53	87.83	82.88	97.48	89.59	89.27	84.44	95.00	89.41	90.00	79.58	88.10	83.68	82.30	78.50	89.04	83.44	82.58	93.52	98.14	95.78	96.05
Bagging	84.73	90.24	87.40	86.57	81.50	88.67	84.93	84.87	85.03	90.00	87.44	88.21	80.36	77.53	78.92	78.85	77.52	84.09	80.67	80.11	91.98	95.67	93.79	94.13
AdaBoost	89.25	87.80	88.52	88.34	85.14	93.71	89.22	89.16	92.74	95.83	94.26	94.70	79.49	83.70	81.45	80.55	82.29	83.74	83.01	83.04	94.31	97.22	95.74	96.00
GBDT	85.15	88.61	86.85	86.18	84.00	92.45	88.02	87.94	86.92	94.16	90.40	91.01	80.43	81.49	80.96	80.38	77.59	81.97	79.72	79.39	90.98	96.60	93.71	94.04
XGBoost	83.84	88.61	86.16	85.33	83.81	91.19	87.34	87.31	85.82	90.83	88.25	88.98	78.62	85.90	82.10	80.73	78.49	81.27	79.86	79.73	91.74	95.98	93.81	94.14
LightGBM	83.33	89.43	86.27	85.31	80.55	91.19	85.54	85.24	85.60	89.16	87.34	88.15	77.35	79.73	78.52	77.65	78.11	81.97	80.00	79.74	90.22	96.91	93.45	93.78
Stacking	89.43	89.43	89.43	89.15	89.88	94.96	92.35	92.45	90.32	93.33	91.80	92.38	81.97	84.14	83.04	82.39	81.94	83.39	82.66	82.69	97.21	96.91	97.06	97.22

Classification performance metrics: Precision (P), Recall (R), F-Measure (F), AUC-ROC (AR).

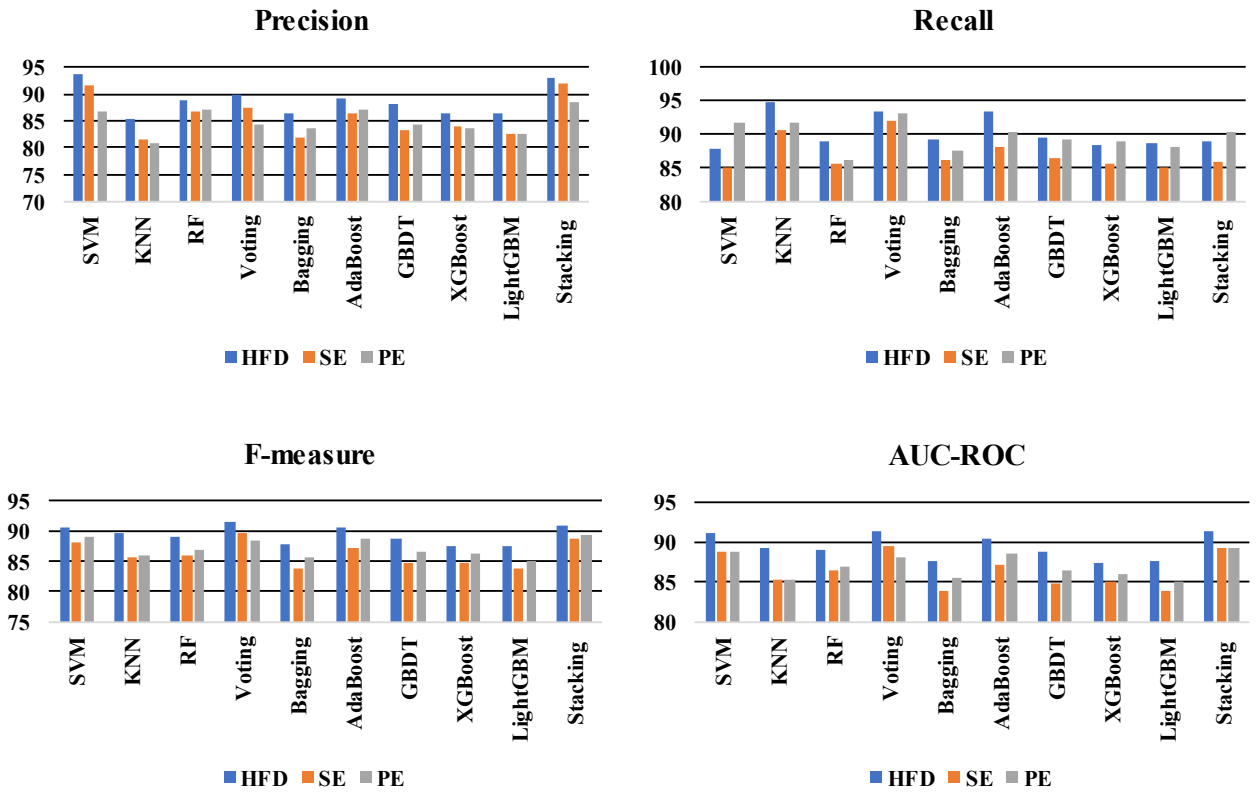


Fig. 7. Average precision, recall, F-measure and AUC-ROC for each classifier considering all electrodes.

Similarly, the probability of choosing (P_{unique}) the data point in the bootstrap sample from the training dataset can also be expressed by the following equation (12).

$$P_{unique} = 1 - \left(1 - \frac{1}{k}\right)^{k} \tag{12}$$

Equation (12) indicates that, $\left(1 - \left(1 - \frac{1}{k}\right)^{k}\right)$ percent of the data points are unique in each bootstrap sample, and other ones are duplicates.

f) Adaptive boosting

Adaptive boosting, usually known as the AdaBoost algorithm, is an ensemble method where multiple weak classifiers are combined to make one robust classifier. Freund and Schapire (Freund & Schapire, 1997) proposed this algorithm in which decision tree stumps are used as a weak classifier sequentially, and the subsequent classifiers correct the wrongly predicted output made by the previous classifier. This process is done by assigning the weights of each sample during each stage of the training period. Each sample weight represents the probability of it being chosen for the training sample by a particular classifier. Each sample weight represents the probability of being selected from training samples by a specific classifier, and probability represents the error rate of each classifier. If the particular classifier misclassifies the class label, the error of the classifier will be increase. On the other hand, if the classifier predicts the data correctly, then the classifier's error decreases. Traditionally, AdaBoost focuses on the misclassification error to boost the performance of the next classifier. It selects a classifier that classifies data with the lowest error rate during the training stage, and these records are chosen to update the weight of training samples for the next classifier.

g) Gradient boosting

Gradient boosting is an algorithm that creates several week learners to make strong learners used in classification and regression problems. In this method, decision trees are added sequentially to improve the errors of the previous trees using the iterative process, and finally, trees are merged to create a strong learner (Friedman, 2002). Indeed, it uses a gradient descent approach to minimize loss when adding new models.

h) Extreme gradient boosting

Extreme gradient boosting, or XGBoost for short, is implemented on the principle of gradient boost, and it is designed in such a way that it runs fast and its performance is also good (T. Chen & Guestrin, 2016). Compared to gradient boosting, XGBoost is preferable since it includes regularization terms that can improve model generalization. Moreover, XGBoost can automatically handle the missing value (sparse data) and work on bigger data than the RAM size. Besides, it uses column subsamples techniques to speed up computation in parallel operation. In addition, XGBoost supports a histogram-based approach for split finding, and it speeds up training and memory utilization (S. Rahman et al., 2020).

i) LightGBM

In 2017, Ke, G et al. proposed a new gradient-boosting decision tree algorithm called LightGBM, combining gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) (Ke et al., 2017). Gradient-based One-Side Sampling is used to estimate the possible splits in the best way. More importantly, it splits the tree leaf-wise. On the other hand, exclusive feature bundling works with numerous features and does not matter whether it is sparse data or not. Therefore, the execution time of LightGBM is faster than XGBoost and GBDT, where an accuracy remains almost the same.

j) Stacking

Stacking is an ensemble method that uses two-level classifiers for classification and regression problems (Wolpert, 1992). In the first level, each classification model is trained using the entire training set to predict the output on the test data set. In the second level, the meta classifier (new classifier) considers all the predictions of first-level classifiers as an input and then predict the final output.

Table 6
Validation (mean ± std. dev.) and test accuracy (%) using DEAP dataset with RFE.

Algorithm	Higuchi Fractal Dimension												
	Sad		Fear		Shamed		Hope		Interest		Excited		Average (test)
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	
SVM	92.41 ± 1.57	92.08	90.07 ± 1.50	91.15	92.39 ± 0.99	95.00	82.75 ± 1.23	86.03	86.91 ± 0.90	84.79	93.35 ± 0.19	93.75	90.47
KNN	90.37 ± 1.42	91.25	89.80 ± 1.96	87.50	91.62 ± 1.31	90.76	81.50 ± 1.15	83.10	85.13 ± 1.66	84.61	93.26 ± 0.57	92.73	88.33
RF	88.80 ± 1.72	89.16	88.58 ± 0.97	88.10	92.74 ± 1.18	92.69	80.70 ± 1.64	84.68	84.35 ± 1.23	84.79	94.16 ± 0.48	94.47	88.98
Voting	92.78 ± 0.95	91.66	90.74 ± 1.36	89.63	93.33 ± 1.17	95.00	83.25 ± 1.37	85.81	86.99 ± 1.22	86.01	95.00 ± 0.44	94.76	90.48
Bagging	89.26 ± 1.86	87.50	87.23 ± 1.20	87.19	89.74 ± 1.85	88.84	79.60 ± 1.25	81.75	82.49 ± 1.22	81.99	91.68 ± 0.86	92.29	86.59
AdaBoost	90.83 ± 1.61	90.00	89.32 ± 0.84	89.02	94.44 ± 0.54	93.07	80.70 ± 1.34	84.90	85.44 ± 1.68	84.09	94.55 ± 0.77	94.76	89.31
GBDT	89.54 ± 2.02	87.91	87.23 ± 0.84	85.97	92.74 ± 1.05	91.92	79.60 ± 1.75	81.75	82.41 ± 1.05	83.21	92.81 ± 0.79	93.02	87.30
XGBoost	89.44 ± 1.93	90.00	86.55 ± 0.89	85.67	91.62 ± 0.96	91.92	78.95 ± 1.87	80.85	83.92 ± 0.99	81.99	91.87 ± 0.87	92.58	87.17
LightGBM	88.43 ± 2.29	87.50	87.23 ± 1.41	86.28	91.11 ± 1.31	90.00	78.35 ± 1.14	81.98	80.39 ± 1.45	80.24	92.68 ± 0.70	93.75	86.63
Stacking	92.78 ± 1.42	91.66	91.01 ± 1.76	90.85	92.74 ± 1.27	96.53	83.50 ± 1.54	86.71	86.87 ± 0.86	85.13	94.84 ± 1.00	95.05	90.99
Algorithm	**Sample Entropy**												
SVM	88.89 ± 1.17	91.25	88.65 ± 1.65	89.93	91.37 ± 1.61	93.84	77.75 ± 0.96	79.95	83.38 ± 1.52	81.64	93.35 ± 0.19	93.75	87.32
KNN	84.07 ± 0.86	84.58	84.59 ± 2.12	84.75	88.97 ± 0.99	90.00	73.25 ± 0.61	73.64	80.23 ± 1.80	81.46	93.26 ± 0.57	92.73	82.89
RF	86.85 ± 1.87	85.83	85.88 ± 1.50	84.14	88.72 ± 2.56	90.76	77.35 ± 1.15	74.77	80.54 ± 1.17	79.72	94.16 ± 0.48	94.47	83.04
Voting	86.57 ± 2.09	87.91	88.38 ± 1.44	88.71	91.71 ± 1.47	93.07	78.05 ± 1.04	79.27	82.68 ± 1.73	84.26	95.00 ± 0.44	94.76	86.64
Bagging	83.89 ± 2.77	82.91	84.05 ± 1.87	82.31	86.50 ± 2.68	87.69	77.55 ± 0.62	75.67	80.43 ± 1.97	79.54	91.68 ± 0.86	92.29	81.62
AdaBoost	87.41 ± 2.53	88.75	88.04 ± 0.79	89.02	90.00 ± 1.98	91.92	79.35 ± 1.02	78.60	81.75 ± 1.66	80.24	94.55 ± 0.77	94.76	85.71
GBDT	83.06 ± 1.48	82.50	83.58 ± 1.49	82.62	87.86 ± 2.44	88.46	75.06 ± 1.67	75.67	78.29 ± 2.28	77.62	92.81 ± 0.79	93.02	81.37
XGBoost	83.43 ± 1.42	80.41	83.04 ± 1.37	84.45	87.18 ± 2.46	88.84	76.05 ± 1.80	75.00	80.04 ± 1.90	77.44	91.87 ± 0.87	92.58	81.23
LightGBM	85.09 ± 1.35	84.58	84.05 ± 1.56	85.67	87.01 ± 2.25	88.46	73.80 ± 1.94	72.74	78.83 ± 1.85	77.79	92.68 ± 0.70	93.75	81.85
Stacking	88.52 ± 1.35	90.83	89.12 ± 1.36	90.24	91.97 ± 1.65	94.61	77.85 ± 1.06	79.27	83.65 ± 2.12	83.56	94.84 ± 1.00	95.05	87.70
Algorithm	**Permutation Entropy**												
SVM	88.43 ± 1.66	87.08	91.69 ± 1.82	89.93	90.60 ± 2.73	91.53	76.95 ± 0.89	77.74	83.15 ± 0.81	83.04	95.68 ± 0.60	95.93	87.54
KNN	85.93 ± 1.91	86.25	84.73 ± 1.23	82.01	87.18 ± 2.40	88.46	78.60 ± 2.08	77.02	81.75 ± 1.44	79.89	89.87 ± 0.48	90.98	84.10
RF	87.13 ± 1.53	86.26	88.72 ± 2.36	84.45	88.97 ± 3.03	88.46	78.75 ± 1.64	81.75	79.34 ± 2.17	78.84	93.52 ± 0.93	93.60	85.56
Voting	90.00 ± 1.59	89.16	89.80 ± 1.20	87.19	90.17 ± 2.28	90.76	80.75 ± 1.41	78.82	84.66 ± 1.20	80.94	94.84 ± 0.14	95.63	87.08
Bagging	86.30 ± 1.67	85.00	85.61 ± 2.09	82.62	87.86 ± 3.23	88.46	79.05 ± 0.91	79.72	79.26 ± 2.63	78.67	91.23 ± 1.20	91.56	84.34
AdaBoost	88.89 ± 2.43	88.33	89.46 ± 1.70	87.50	90.34 ± 2.67	90.38	79.85 ± 2.17	78.60	80.58 ± 0.79	78.14	93.65 ± 0.98	94.04	86.17
GBDT	86.02 ± 2.04	83.75	87.03 ± 1.38	81.09	87.69 ± 2.03	86.53	77.30 ± 1.30	78.60	78.99 ± 1.12	79.19	92.48 ± 1.15	92.58	83.62
XGBoost	85.37 ± 2.35	85.00	84.73 ± 2.10	82.01	87.86 ± 3.18	88.07	77.10 ± 2.64	77.92	77.01 ± 1.89	76.04	91.48 ± 1.18	91.71	83.46
LightGBM	85.46 ± 2.16	82.50	86.28 ± 2.15	81.70	88.80 ± 2.84	87.69	75.90 ± 1.76	76.57	77.83 ± 1.06	77.97	90.87 ± 0.89	90.84	82.88
Stacking	90.28 ± 1.28	90.00	90.95 ± 0.94	90.54	90.77 ± 2.46	91.53	80.70 ± 0.99	79.05	84.66 ± 1.33	82.16	96.00 ± 0.45	96.36	88.27

3.7. Statistical analysis

In this paper, the Combined 5 × 2 cv F test was used to compare the performance of the two models (Alpaydin, 1999). According to this test, the F-statistic is computed as follows:

$$f = \frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\left(p_i^j\right)^2}{2\sum_{i=1}^{5}s_i^2} \tag{13}$$

where p is the performance of the classifier and s is the variance of the difference between two algorithms. Furthermore, the p-value is calculated using the f statistic and compared to a previously determined

Table 7
Precision, recall, F-measure and AUC-ROC (%) using DEAP dataset with RFE.

Algorithm	Higuchi Fractal Dimension																							
	Sad				Fear				Shamed				Hope				Interest				Excited			
	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR
SVM	91.66	89.43	90.53	90.44	87.42	96.22	91.61	91.60	92.00	95.83	93.87	94.34	82.71	88.54	85.53	84.59	85.17	87.27	86.21	86.20	89.57	98.14	93.66	93.99
KNN	87.59	91.86	89.68	89.09	78.57	96.85	86.70	86.00	87.40	98.33	92.54	93.09	79.84	92.51	85.71	84.04	81.52	90.45	85.76	85.19	89.20	96.91	92.89	93.23
RF	88.23	85.36	86.77	86.70	84.48	92.45	88.28	88.23	91.59	90.83	91.21	91.84	83.03	81.93	82.48	82.21	80.53	84.80	82.61	82.36	94.57	96.91	95.73	95.98
Voting	91.80	91.05	91.42	91.25	83.78	97.48	90.11	89.86	93.44	95.00	94.21	94.64	84.51	88.98	86.69	85.96	84.28	89.04	86.59	86.39	73.90	83.02	78.19	78.46
Bagging	87.50	85.36	86.41	86.27	82.65	89.93	86.14	86.09	86.61	91.66	89.06	89.76	79.07	83.25	81.11	80.10	78.59	83.03	80.75	80.44	92.92	97.22	95.02	95.31
AdaBoost	90.83	88.61	89.71	89.60	84.97	92.45	88.55	88.53	92.62	94.16	93.38	93.86	81.97	84.14	83.04	82.39	82.66	87.63	85.07	84.81	90.40	95.98	93.11	93.46
GBDT	87.39	84.55	85.95	85.86	83.05	92.45	87.50	87.35	88.09	92.50	90.24	90.89	78.86	85.46	82.02	80.74	82.43	86.21	84.28	84.11	93.00	98.45	95.65	95.93
XGBoost	88.13	84.55	86.30	86.29	82.02	91.82	86.64	86.64	86.40	90.00	88.16	88.92	78.15	81.93	80.00	78.98	79.25	81.97	80.83	80.78	90.46	96.60	93.43	93.76
LightGBM	86.55	83.73	85.21	85.03	83.05	92.45	87.50	87.35	93.27	92.50	92.88	93.39	80.16	83.70	81.89	81.02	78.21	83.74	80.88	80.54	89.88	95.98	92.83	93.18
Stacking	92.37	88.61	90.45	90.46	89.69	93.08	91.35	91.51	96.58	94.16	95.35	95.65	87.21	84.14	85.65	85.61	87.05	85.51	86.27	86.52	95.37	95.37	95.37	95.62
Algorithm	**Sample Entropy**																							
SVM	92.92	85.36	88.98	89.26	86.98	92.45	89.63	89.71	94.64	88.33	91.37	92.02	82.32	77.97	80.00	80.23	82.57	83.74	83.15	83.22	94.11	98.76	96.38	96.63
KNN	85.00	82.92	83.95	83.77	79.53	85.53	82.42	82.41	88.70	91.66	90.16	90.83	73.23	86.78	79.43	76.80	73.33	89.39	80.57	78.78	83.42	96.29	89.39	89.63
RF	86.99	86.99	86.99	86.65	82.82	84.90	83.85	84.16	88.98	87.50	88.23	89.10	78.60	74.44	76.47	76.62	79.93	83.03	81.45	81.31	89.69	91.35	90.51	91.00
Voting	87.06	87.80	87.44	87.06	84.09	93.08	88.35	88.25	91.59	90.83	91.21	91.84	77.59	82.37	79.91	78.74	79.09	86.92	82.82	82.21	94.34	97.83	96.06	96.30
Bagging	85.48	86.17	85.82	85.39	83.64	83.64	83.64	84.13	83.33	87.50	85.36	85.25	77.77	77.09	77.43	77.02	75.81	81.97	78.77	78.18	84.87	93.51	88.98	89.34
AdaBoost	90.90	89.43	90.16	90.01	85.02	89.30	87.11	87.25	90.83	90.83	90.83	91.48	77.77	80.17	78.95	78.10	79.41	85.86	82.51	82.03	92.03	96.29	94.11	94.43
GBDT	86.77	85.36	86.06	85.84	81.65	86.79	84.14	84.22	85.36	87.50	86.41	87.32	78.12	77.09	77.60	77.25	76.50	85.15	80.60	79.77	85.26	92.90	88.92	89.30
XGBoost	87.70	86.99	87.34	87.08	81.54	86.16	83.79	83.91	83.59	89.16	86.29	87.08	78.07	78.41	78.24	77.68	75.49	81.62	78.43	77.83	87.10	93.82	90.34	90.73
LightGBM	83.87	84.55	84.21	83.72	81.17	86.79	83.89	83.92	84.92	89.16	86.99	87.79	74.77	74.44	74.61	74.09	76.87	83.39	80.00	79.41	87.14	94.13	90.50	90.88
Stacking	92.79	83.73	88.03	88.45	88.95	91.19	90.06	90.27	97.24	88.33	92.57	93.09	82.69	75.77	79.08	79.59	84.75	80.56	82.60	83.18	96.91	96.91	96.91	97.08
Algorithm	**Permutation Entropy**																							
SVM	80.45	86.99	83.59	82.38	89.30	89.30	89.30	89.62	88.79	85.83	87.28	88.27	84.07	74.44	78.97	79.85	85.37	76.32	80.59	81.76	95.03	94.44	94.73	95.02
KNN	85.95	84.55	85.24	85.01	74.14	95.59	83.51	82.11	79.02	94.16	85.93	86.36	73.89	88.54	80.56	77.91	76.04	89.75	82.33	81.03	84.45	97.22	90.38	90.64
RF	87.28	83.73	85.47	85.45	86.07	85.53	85.80	86.25	84.00	87.50	85.71	86.60	82.43	80.61	81.51	81.32	79.02	79.85	79.43	79.54	92.74	94.75	93.74	94.07
Voting	83.84	88.61	86.16	85.33	83.24	93.71	88.16	87.97	85.93	91.66	88.70	89.40	79.35	86.34	82.70	81.42	79.73	84.80	82.19	81.84	94.15	94.44	94.29	94.61
Bagging	85.21	79.67	82.35	82.57	83.03	86.16	84.56	84.79	81.53	88.33	84.80	85.59	79.74	83.25	81.46	80.57	75.96	83.74	79.66	78.89	87.74	95.06	91.25	91.62
AdaBoost	87.60	86.17	86.88	86.67	84.52	89.30	86.85	86.96	84.73	92.50	88.44	89.10	81.58	85.90	83.69	82.81	77.25	81.62	79.38	79.04	92.08	96.91	94.43	94.74
GBDT	83.06	83.73	83.40	82.89	80.22	89.30	84.52	84.29	84.37	90.00	87.09	87.85	77.82	81.93	79.82	78.75	75.40	82.33	78.71	78.01	88.76	95.06	91.80	92.17
XGBoost	84.12	86.17	85.14	84.54	83.52	89.30	86.32	86.37	83.20	86.66	84.89	85.83	78.29	81.05	79.65	78.77	75.33	79.85	77.53	77.12	87.57	93.51	90.44	90.85
LightGBM	83.60	82.92	83.26	82.91	80.45	88.05	84.08	83.96	80.76	87.50	84.00	84.21	77.39	78.41	77.89	77.22	73.98	77.38	75.64	75.37	86.88	91.97	89.35	89.80
Stacking	84.42	83.73	84.08	83.75	88.81	89.93	89.37	89.64	89.56	85.83	87.65	88.63	83.49	75.77	79.44	80.05	85.09	76.67	80.66	81.76	96.47	92.90	94.65	94.93

Classification performance metrics: Precision (P), Recall (R), F-Measure (F), AUC-ROC (AR).

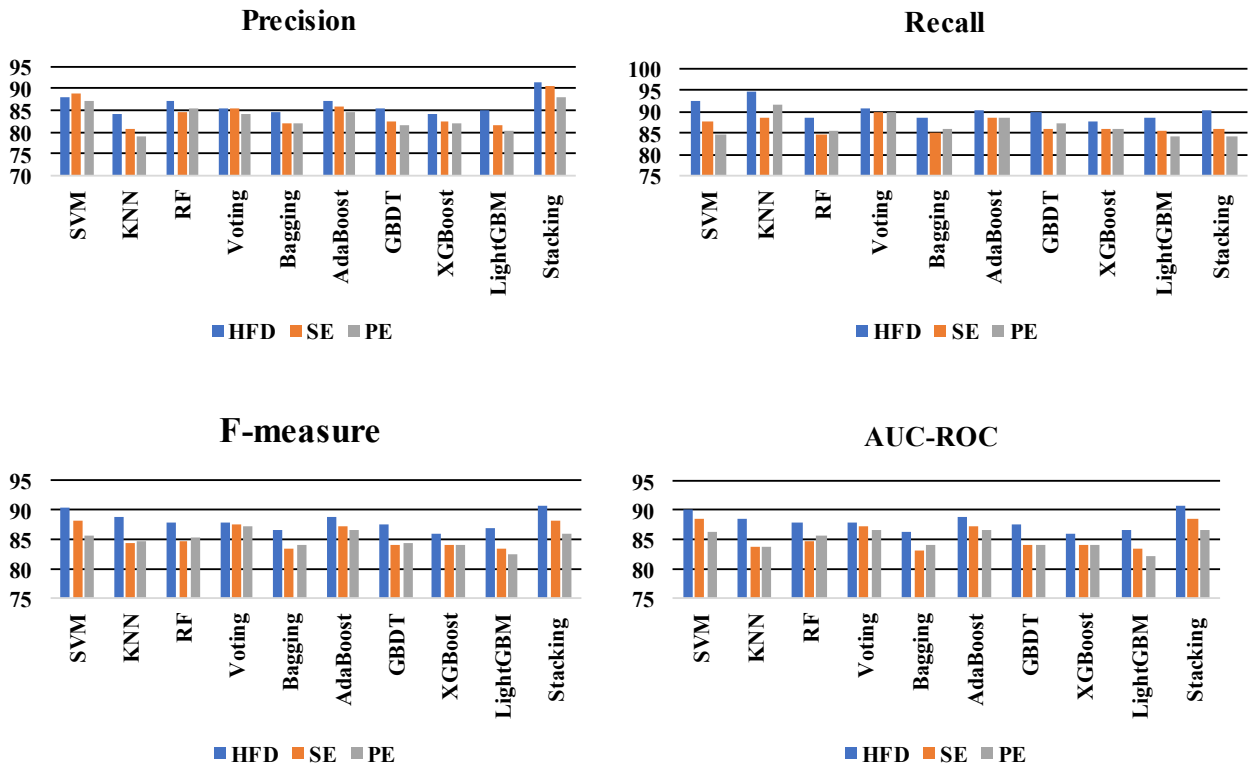


Fig. 8. Average precision, recall, F-measure and AUC-ROC for each classifier after electrode reduction.

significance level ($\alpha = 0.05$). If the p-value is less than α, we reject the null hypothesis and agree that the two models are significantly different. Similarly, we employed the non-parametric statistic Friedman test with the appropriate post-hoc tests to compare multiple classifiers over several data sets (Demšar, 2006). In this test, according to the null hypothesis, the performance of all algorithms is the same. We can reject the null hypothesis only if $p < 0.05$.

4. Experimental setup

This paper focuses on evaluating six basic emotions using the EEG signal of the DEAP dataset. Similarly, another dataset named AMIGOS is used to validate the model performance. The experiments are performed using a single computer (Intel R Core (TM) i5 − 6200U CPU @ 2.30 GHz, 8.0 GB RAM) with a windows 10 operating system. The EEG features are calculated using MATLAB R2020a software while hyperparameter tuning and classification performance are stimulated in Python programming language through Jupyter notebook.

4.1. Splitting the dataset

In this experiment, we have set 10% of the class labels and their corresponding EEG features as test data and the rest 90% of the class labels and EEG features as training data. Then, we have used the shuffle-split cross-validation technique to train the model effectively. In the cross-validation technique, 90% of previous training data have further been splitted into five groups (splitting iteration), and each group consists of a training set of 90% and a test set of 10%. Fig. 5 illustrates the shuffle-split cross-validation technique.

4.2. Parameter tunning

Hyperparameters in machine learning algorithms help us improve the algorithm's performance on a specific dataset. However, not all hyperparameters of an algorithm are equally important. Nevertheless, some parameters affect the behavior of classification models and can deliver the best performance by selecting suitable parameters. The tuning process becomes more difficult and slower as an algorithm's hyperparameters increase. Therefore, it is essential to find a minimum subset of model hyperparameters to optimize the model performance. Random search is a strategy in which only a few combinations of hyperparameters are randomly selected from the entire combinations and used to find the optimum solution. Fig. 6 illustrates the process of hyperparameter tunning using the randomized grid search cross-validation method. During the training session, we tuned the effective hyperparameters of each classifier for each emotion using a randomized grid search cross-validation technique that is more efficient than the grid search method (Bergstra & Bengio, 2012). Then the optimal parameters of each classifier have been used for the final prediction of target emotions that can give better results. Table 3 represents all key hyperparameters of an individual classifier for binary classification.

5. Results and discussion

This paper extracted three nonlinear features to classify six discrete emotions using the DEAP and AMIGOS dataset. Total ten classification methods, including eight ensemble methods (SVM, KNN, RF, Voting, Bagging, AdaBoost, GBDT, XGBoost, LightGBM, stacking) were used to classify emotions along with their effective hyperparameters to increase the performance of the classifiers. Each EEG feature and their corresponding emotional states were divided into training and test data in which 90% of the data was used as training data, and the rest (10%) were used as test data. All the models were trained using shuffle-split cross-validation technique. Firstly, we used thirty-two electrodes for each of the experiments, and then all the experiments were done by reducing the number of electrodes. A reduction in the number of electrodes was made using the recursive feature elimination (REF) method.

a. Classification performance for DEAP dataset

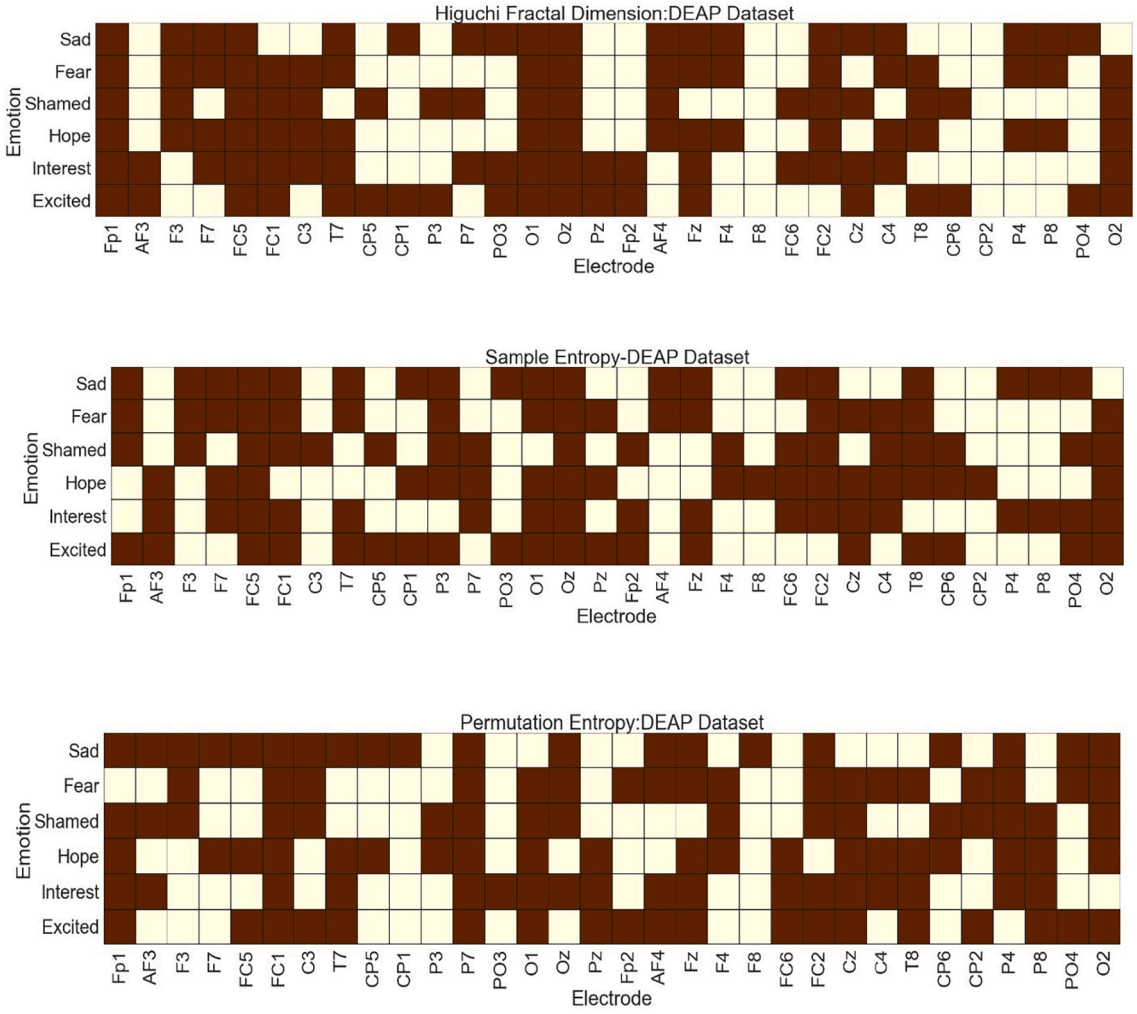


Fig. 9. Heat map of the selected channels for each emotion using three non-linear features on DEAP dataset. The red color in the square box indicates the presence of electrodes for each emotion.

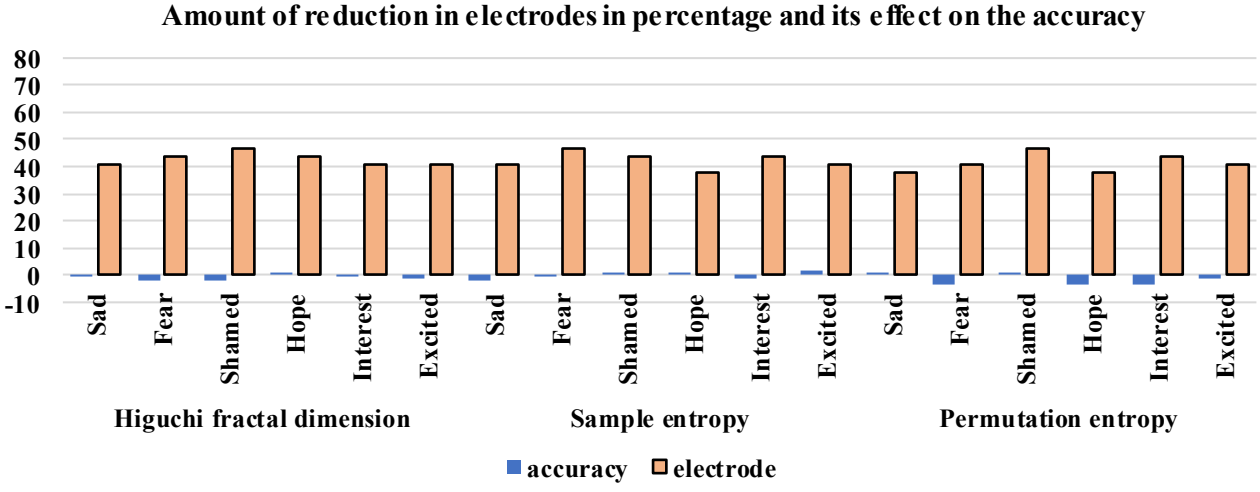


Fig. 10. Amount of reduction of electrodes (%) and changes of overall accuracy (%).

In this section, the performance of each classifier has been evaluated based on accuracy, precision, recall, F-measure, and AUC-ROC. The performance of the classifiers has been shown considering all electrodes and an optimal number of electrodes related to emotions.

i) Without electrode reduction

The outcomes of the experiments were shown in Table 4 and Table 5.

Table 8
Validation (mean ± std. dev.) and test accuracy (%) using AMIGOS dataset without RFE.

Algorithm	Higuchi Fractal Dimension												
	Sad		Fear		Shamed		Hope		Interest		Excited		Average (test)
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	
SVM	89.74 ± 1.66	85.93	93.74 ± 1.96	97.27	88.52 ± 3.59	85.00	93.33 ± 2.50	95.28	93.43 ± 1.29	97.91	100.00	100.00	93.57
KNN	88.00 ± 1.00	84.26	91.72 ± 1.48	97.27	88.15 ± 2.77	88.33	92.14 ± 2.25	92.30	97.59 ± 1.69	99.58	98.76 ± 0.59	99.20	93.49
RF	87.75 ± 2.00	80.89	89.29 ± 3.48	92.72	81.48 ± 5.62	83.33	89.23 ± 2.94	92.30	97.69 ± 1.06	100.00	99.73 ± 0.22	99.60	91.47
Voting	92.00 ± 1.50	89.88	94.14 ± 2.34	98.18	89.63 ± 3.99	86.66	93.85 ± 1.83	96.15	97.69 ± 1.31	98.75	49.60 ± 2.72	49.60	86.54
Bagging	86.25 ± 3.95	79.77	91.31 ± 1.87	93.63	81.48 ± 4.68	90.00	89.06 ± 2.72	84.61	96.57 ± 1.57	99.58	98.40 ± 0.36	99.20	91.13
AdaBoost	90.75 ± 1.00	87.64	92.73 ± 2.42	97.27	86.67 ± 3.95	86.66	90.94 ± 2.45	96.15	99.35 ± 0.47	100.00	100.00	100.00	94.62
GBDT	87.50 ± 1.37	79.77	90.10 ± 4.01	96.36	81.85 ± 5.02	83.33	90.09 ± 2.57	91.53	98.43 ± 1.23	100.00	99.38 ± 0.45	99.60	91.77
XGBoost	89.25 ± 2.03	85.39	91.11 ± 2.25	94.54	81.11 ± 3.59	83.33	88.55 ± 2.89	93.07	96.48 ± 1.36	98.75	98.93 ± 0.45	99.60	92.45
LightGBM	88.50 ± 1.66	84.26	89.90 ± 2.63	94.54	81.85 ± 6.35	78.33	88.72 ± 2.72	90.76	97.87 ± 1.36	100.00	97.87 ± 0.52	98.80	91.12
Stacking	91.00 ± 2.00	89.88	94.14 ± 2.34	97.27	88.52 ± 4.44	88.33	93.68 ± 2.27	95.28	97.41 ± 1.67	99.58	50.40 ± 2.72	50.40	86.79
Algorithm	**Sample Entropy**												
SVM	84.25 ± 1.70	84.26	90.10 ± 3.34	90.00	79.63 ± 4.97	80.00	91.79 ± 1.49	87.69	99.26 ± 1.04	99.16	99.73 ± 0.36	99.60	90.12
KNN	81.00 ± 1.84	85.39	91.11 ± 1.74	89.09	82.22 ± 2.77	80.00	89.40 ± 2.33	92.30	96.39 ± 1.22	97.91	99.11 ± 0.56	97.60	90.38
RF	81.00 ± 2.15	79.77	86.67 ± 1.85	93.63	80.74 ± 3.43	80.00	88.72 ± 1.98	90.00	97.59 ± 0.54	99.16	99.20 ± 0.59	98.80	90.23
Voting	83.75 ± 2.37	85.39	91.11 ± 2.81	92.72	84.44 ± 2.51	81.66	91.79 ± 2.51	91.53	98.80 ± 0.86	100.00	80.09 ± 2.43	100.00	91.88
Bagging	81.00 ± 1.46	79.77	86.26 ± 1.51	93.63	75.93 ± 4.83	76.66	86.32 ± 4.08	88.46	97.59 ± 0.94	98.33	97.96 ± 0.82	98.40	89.21
AdaBoost	85.50 ± 2.81	82.02	88.89 ± 1.69	95.45	83.33 ± 2.34	81.66	91.79 ± 2.45	92.30	98.61 ± 0.72	100.00	99.73 ± 0.36	99.60	91.84
GBDT	83.50 ± 2.78	79.77	86.46 ± 2.90	92.72	79.63 ± 3.10	81.66	87.69 ± 3.09	89.23	97.87 ± 0.91	99.16	99.82 ± 0.22	98.40	90.16
XGBoost	81.75 ± 4.23	82.02	84.65 ± 2.96	93.63	77.78 ± 5.62	73.33	87.52 ± 1.92	87.69	97.13 ± 1.50	97.91	99.38 ± 0.53	97.20	88.63
LightGBM	82.50 ± 2.85	82.02	86.46 ± 2.97	90.90	78.89 ± 3.63	78.33	88.89 ± 3.42	90.00	97.31 ± 1.26	97.91	99.11 ± 0.56	96.00	89.19
Stacking	84.50 ± 2.03	87.64	90.71 ± 3.16	90.90	81.11 ± 2.46	83.33	92.65 ± 0.87	90.00	99.26 ± 1.04	99.16	79.82 ± 24.53	100.00	91.84
Algorithm	**Permutation Entropy**												
SVM	81.25 ± 2.62	76.40	89.90 ± 1.69	95.45	78.15 ± 6.35	81.66	86.84 ± 1.39	92.30	98.89 ± 0.86	98.75	99.91 ± 0.18	100.00	90.76
KNN	77.25 ± 2.42	70.78	89.09 ± 2.06	93.63	78.89 ± 4.16	83.33	82.56 ± 2.33	91.53	95.28 ± 1.45	96.25	97.07 ± 0.92	98.00	88.92
RF	78.75 ± 1.37	79.77	86.06 ± 2.16	90.90	81.85 ± 4.60	76.66	82.56 ± 2.94	90.76	96.94 ± 0.69	97.91	99.02 ± 0.52	98.40	89.07
Voting	80.75 ± 1.00	75.28	92.12 ± 1.96	94.54	80.00 ± 3.78	86.66	83.76 ± 2.23	90.76	98.52 ± 0.54	98.33	68.89 ± 2.49	49.60	82.53
Bagging	79.25 ± 3.76	83.14	87.47 ± 2.44	89.09	79.63 ± 5.11	73.33	82.91 ± 3.94	90.00	95.28 ± 1.69	96.66	97.24 ± 1.21	98.00	88.37
AdaBoost	80.25 ± 2.00	79.77	90.71 ± 0.76	93.63	82.22 ± 2.51	76.66	86.50 ± 2.32	93.84	98.43 ± 1.04	99.16	99.73 ± 0.36	99.20	90.38
GBDT	75.50 ± 2.03	77.52	86.67 ± 2.06	89.09	81.11 ± 5.67	76.66	82.91 ± 1.62	90.00	96.76 ± 0.97	97.08	99.56 ± 0.28	99.60	88.33
XGBoost	78.25 ± 3.22	78.65	87.07 ± 1.18	89.09	77.78 ± 2.62	70.00	84.10 ± 2.79	89.23	96.11 ± 1.39	97.08	99.02 ± 0.44	98.80	87.14
LightGBM	76.00 ± 2.29	76.40	87.07 ± 1.18	90.90	77.78 ± 2.62	76.66	81.71 ± 1.76	89.23	95.56 ± 1.89	97.50	98.67 ± 0.40	98.80	88.25
Stacking	80.50 ± 3.41	75.28	91.92 ± 1.43	96.36	78.52 ± 4.48	83.33	86.67 ± 2.20	93.07	99.07 ± 0.51	99.16	70.84 ± 23.91	50.40	82.93

The table illustrates the overall performance of each classifier using the Higuchi fractal dimension, sample entropy, and permutation entropy, respectively. The highlighted number in the table indicates the best performance of each algorithm. From Table 4, it was seen that the stacking ensemble method showed the highest performance. Moreover, the voting ensemble method showed satisfactory performance in categorizing emotions like sad, hope, and shamed using HFD, SE, and PE, respectively. In addition, the SVM classifier showed the superior performance in analyzing negative emotions like fear using HFD. Considering all emotions, it is clear that the highest average accuracy has been achieved using the stacking classifier. The stacking classifier achieved an average accuracy of 91.53%, 89.40%, and 87.32% for HFD, SE, and PE respectively. Similarly, it has also been proven that the Higuchi fractal dimension was the most significant feature among the three features

Table 9
Precision, recall, F-measure and AUC-ROC (%) using AMIGOS dataset without RFE.

Algorithm	Higuchi Fractal Dimension																							
	Sad				Fear				Shamed				Hope				Interest				Excited			
	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR
SVM	79.16	92.68	85.93	85.92	100	90.16	94.82	95.08	89.65	89.65	89.65	89.98	96.15	93.75	94.93	93.87	100	100	100	100	100	100	100	100
KNN	84.44	92.68	88.37	89.04	98.27	93.44	95.79	95.70	87.09	93.10	90.00	90.10	93.90	96.25	95.06	93.12	99.20	99.20	99.20	99.16	98.43	100	99.21	99.19
RF	85.71	87.80	86.74	87.65	92.06	95.08	93.54	92.43	86.66	89.65	88.13	88.37	96.05	91.25	93.58	92.62	100	98.40	99.19	99.20	100	100	100	100
Voting	80.85	92.68	86.36	86.96	100	98.36	99.17	99.18	87.09	93.10	90.00	90.10	96.29	97.50	96.89	95.75	100	100	100	100	50.53	52.43	47.83	50
Bagging	80.95	82.92	81.92	83.13	93.33	91.80	92.56	91.82	83.87	89.65	86.66	86.76	90.24	92.50	91.35	88.25	100	100	100	100	98.43	100	99.21	99.19
AdaBoost	82.22	90.24	86.04	86.78	95.23	98.36	96.77	96.11	84.84	96.55	90.32	90.21	96.20	95.00	95.59	94.50	100	98.40	99.19	99.20	100	100	100	100
GBDT	77.08	90.24	83.14	83.66	93.65	96.72	95.16	94.27	78.78	89.65	83.87	83.53	92.59	93.75	93.16	90.87	100	100	100	100	99.21	100	99.60	99.59
XGBoost	81.81	87.80	84.70	85.56	90.76	96.72	93.65	92.23	78.78	89.65	81.25	80.31	91.46	93.75	92.59	89.87	100	100	100	100	99.21	100	99.60	99.59
LightGBM	81.81	87.80	84.70	85.56	93.84	100	96.82	95.91	74.28	89.65	81.25	80.31	89.15	92.50	90.79	87.25	100	100	100	100	99.21	100	99.60	99.59
Stacking	81.25	95.12	87.64	88.18	100	98.36	99.17	99.18	87.09	93.10	90.00	90.10	96.15	93.75	94.93	93.87	100	98.40	99.19	99.20	0.504	100	67.02	50.00
Algorithm	**Sample Entropy**																							
SVM	79.06	82.92	80.95	82.08	87.50	91.80	89.59	87.73	91.30	72.41	80.76	82.98	97.40	93.75	95.54	94.87	100	100	100	100	100	99.20	99.60	99.60
KNN	81.08	73.17	76.92	79.29	95.16	96.72	95.93	95.29	75.00	82.75	78.68	78.47	89.77	98.75	94.04	90.37	97.65	100	98.81	98.69	95.45	100	97.67	97.58
RF	78.04	78.04	78.04	79.64	93.10	88.52	90.75	90.18	75.75	86.20	80.64	80.20	93.67	92.50	93.08	91.25	100	99.20	99.59	99.60	98.43	100	99.21	99.19
Voting	79.54	85.36	82.35	83.30	95.08	95.08	95.08	94.47	79.41	93.10	85.71	85.26	95.12	97.50	96.29	94.75	100	100	100	100	100	100	100	100
Bagging	80.48	80.48	80.48	81.91	91.66	90.16	90.90	89.97	71.42	86.20	78.12	76.97	91.35	92.50	91.92	89.25	99.19	98.40	98.79	98.76	96.18	100	98.05	97.98
AdaBoost	82.05	78.04	80.00	81.73	93.44	93.44	93.44	92.63	83.33	86.20	84.74	85.03	95.12	97.50	96.29	94.75	100	99.20	99.59	99.60	99.21	100	99.60	99.59
GBDT	80.48	80.48	80.48	81.91	90.62	95.08	92.79	91.41	77.41	82.75	79.99	80.08	93.75	93.75	93.75	91.87	100	99.20	99.59	99.60	97.67	100	98.82	98.79
XGBoost	81.57	75.60	78.48	80.51	86.56	95.08	90.62	88.35	70.58	82.75	76.19	75.25	89.87	88.75	89.30	86.37	96.09	98.40	97.23	97.02	94.73	100	97.29	97.17
LightGBM	82.05	78.04	80.00	81.73	87.87	95.08	91.33	89.37	73.52	86.20	79.36	78.58	93.67	92.50	93.08	91.25	96.85	98.40	97.61	97.46	94.73	100	97.29	97.17
Stacking	79.06	82.92	80.95	82.08	95.00	93.44	94.21	93.66	83.87	89.65	86.66	86.76	96.20	95.00	95.59	94.50	100	100	100	100	100	99.20	99.60	99.60
Algorithm	**Permutation Entropy**																							
SVM	67.34	80.48	73.33	73.57	92.06	95.08	93.54	92.43	76.66	79.31	77.96	78.36	94.66	88.75	91.63	90.37	98.41	99.20	98.80	98.73	100	100	100	100
KNN	68.29	68.29	68.29	70.60	90.76	96.72	93.65	92.23	77.77	72.41	75.00	76.52	88.63	97.50	92.85	88.75	93.28	100	96.52	96.08	96.18	100	98.05	97.98
RF	85.00	82.92	83.95	85.21	91.37	86.88	89.07	88.34	71.87	79.31	75.40	75.13	93.50	90.00	91.71	90.00	98.41	99.20	98.80	98.73	96.92	100	98.43	98.38
Voting	66.66	82.92	73.91	73.75	93.75	96.72	95.16	94.27	76.47	89.65	82.53	81.92	89.41	95.00	92.12	88.50	99.20	99.20	99.20	99.16	50.53	52.43	47.83	50.00
Bagging	87.17	82.92	85.00	86.25	91.80	91.80	91.80	90.79	75.00	72.41	73.68	74.91	91.25	91.25	91.25	88.62	94.57	97.60	96.06	95.75	96.18	100	98.05	97.98
AdaBoost	74.46	85.36	79.54	80.18	93.44	93.44	93.44	92.63	67.64	79.31	73.01	71.91	93.75	93.75	93.75	91.87	96.80	96.80	96.80	96.66	98.43	100	99.21	99.19
GBDT	72.72	78.04	75.29	76.52	91.93	93.44	92.68	91.61	75.86	75.86	75.86	76.64	92.20	88.75	90.44	88.37	96.09	98.40	97.23	97.02	99.21	100	99.60	99.59
XGBoost	73.91	82.92	78.16	78.77	85.71	88.52	87.09	85.07	66.66	68.96	67.79	68.53	92.40	91.25	91.82	89.62	98.40	98.40	98.40	98.33	97.67	100	98.82	98.79
LightGBM	74.46	82.92	78.16	78.96	87.30	90.16	88.70	86.91	72.41	72.41	72.41	73.30	91.46	93.75	92.59	89.87	96.09	98.40	97.23	97.02	97.67	100	98.82	98.79
Stacking	71.73	80.48	75.86	76.70	96.66	95.08	95.86	95.50	80.00	82.75	81.35	81.70	92.40	91.25	91.82	89.62	98.41	99.20	98.80	98.73	50.4	100	67.02	50.00

Classification performance metrics: Precision (P), Recall (R), F-Measure (F), AUC-ROC (AR).

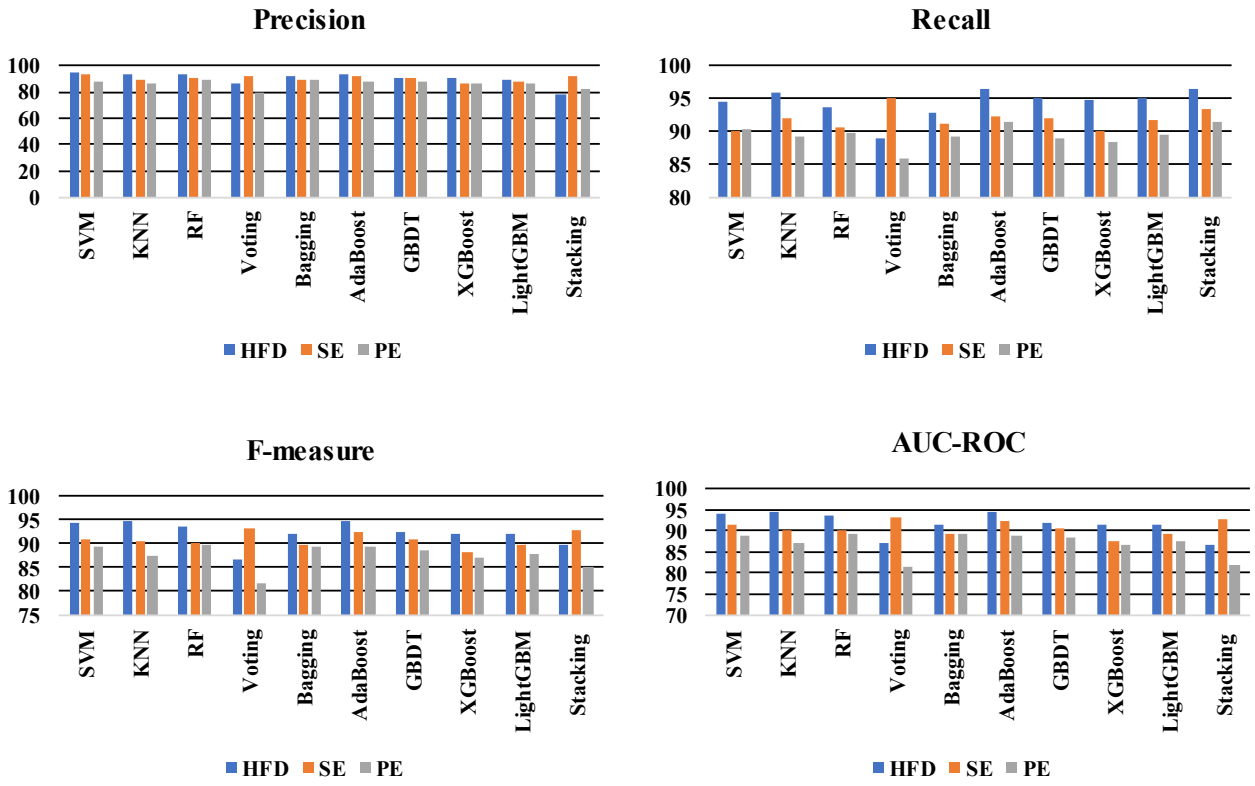


Fig. 11. Average precision, recall, F-measure and AUC-ROC for each classifier considering all electrodes.

because it showed more accuracy than the other two features. For instance, we obtained 91.53% accuracy considering all the electrodes.

Moreover, precision, recall, F-measure, and AUC-ROC have been evaluated for the same dataset and experimental setting. The comparative analysis of each classifier based on different performance metrics (average) is shown in Fig. 7. The SVM classifier achieved the highest 93.66% precision using the HFD feature. The stacking classifier achieved 91.84% and 88.46% precision for SE and PE features. Similarly, KNN obtained the highest 94.79% recall for HFD, and the voting classifier achieved 92.13% and 93.14% recall for SE and PE, respectively. In the case of the F-measure, voting gained 91.52% and 89.57% for HFD and SE features. Stacking achieved an average of 89.39% F-measure for the PE feature. In the case of performance metrics, the HFD feature outperformed the other two features. Therefore, HFD ensures the consistency of HFD performance compared to the other two features.

ii) Electrode reduction

Reducing the number of electrodes to make the system easier is one of the essential issues in emotion analysis. Therefore, the recursive feature elimination method was used as a selector of relevant electrodes to each emotion. The results of each classifier are shown in Table 6 and Table 7 respectively. The stacking classifier achieved an average accuracy of 90.99%, 87.70%, and 88.27% for HFD, SE, and PE respectively. The comparison of average precision, recall, F-measure, and AUC-ROC of each classifier are shown in Fig. 8. The stacking classifier achieved 91.38%, 90.56%, and 87.97% precision for HFD, SE, and PE. Similarly, KNN reached 94.49% and 91.64% recall for HFD and PE features. The voting classifier obtained an average recall of 89.81% for the SE feature. To obtain the F-measure, the stacking classifier performed well using HFD and whose value was 90.74%, whereas SVM and voting achieved 88.25% and 87.03% for SE and PE, respectively. In the case of AUC-ROC, the stacking algorithm gained 90.90% and 88.61% using HFD and SE, whereas the voting classifier obtained 86.76% for the PE feature.

A heat map of the selected channels for each emotion using three non-linear features is shown in Fig. 9. We can see that, for shame, the total number of electrodes was 18 and which is about 46.88% less than the total number of electrodes. Then, those electrodes were used for the analysis of each emotion. It was observed that when the number of electrodes was reduced, the accuracy for each emotional state decreased, but the rate of reduction was not >2% as compared to the previous accuracy. The bar chart (see Fig. 10) illustrates the reduction rate of electrodes and its effect on the overall accuracy. It was seen that the average accuracy for each emotion little bit changed after reducing the number of electrodes. Overall, the change in average accuracy for each emotion using HFD was lower than SE and PE. In terms of HFD, the average accuracy decreased for four emotions (sad, fear, shamed, interest, excited) while it increased to 0.62% for hope. In contrast, for all the emotions number of electrodes decreased to 40.63%, 43.75%, 46.88%, 43.75%, 40.63%, and 40.63% respectively. Similarly, in terms of permutation entropy and sample entropy, average accuracy was changed for changing the number of electrodes. For interest, the change of average accuracy was greater when SE and PE were used, but in terms of HFD, it was approximately equal to the previous accuracy.

b. Classification performance for AMIGOS dataset

i) Without electrode reduction

This paper used another publicly available database, named AMIGOS, to check the model performance. Like the DEAP dataset, all the experiments have been conducted by analyzing all electrodes and the selected electrodes. The outcomes of each algorithm are shown in Table 8 and Table 9. AdaBoost achieved an average accuracy of 94.62% for HFD. The voting and SVM achieved 91.88% and 90.76% accuracy for SE and PE, respectively. It was seen that the HFD feature outperformed the other two features in terms of accuracy. Moreover, The RF classifier achieved the highest 89.51% precision using the PE feature. The SVM classifier achieved 94.16% and 92.54% precision for HFD and SE features. Similarly, voting classifier obtained 95.17% recall for SE, and the stacking classifier achieved 96.46% and 91.46% recall for SE and PE,

Table 10
Validation (mean ± std. dev.) and test accuracy (%) using AMIGOS dataset with RFE.

Algorithm	Higuchi Fractal Dimension												
	Sad		Fear		Shamed		Hope		Interest		Excited		Average (test)
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	
SVM	82.75 ± 3.20	80.89	84.85 ± 2.12	87.27	82.22 ± 3.81	81.66	68.15 ± 6.77	70.00	98.06 ± 0.74	97.08	99.64 ± 0.33	100.00	86.15
KNN	81.00 ± 4.50	76.40	87.27 ± 1.03	91.81	80.74 ± 4.77	73.33	75.56 ± 3.95	73.33	96.20 ± 0.61	97.08	97.42 ± 0.59	98.00	84.99
RF	77.75 ± 3.10	80.89	86.67 ± 2.74	89.09	81.11 ± 3.59	78.33	76.30 ± 7.72	81.66	96.94 ± 1.45	96.25	98.49 ± 0.60	98.80	87.50
Voting	82.25 ± 3.66	82.02	88.89 ± 1.28	91.81	85.19 ± 5.11	71.66	76.30 ± 2.72	73.33	98.43 ± 0.69	97.91	99.56 ± 0.40	99.60	86.06
Bagging	78.00 ± 1.87	79.77	86.87 ± 1.43	89.09	81.48 ± 3.31	78.33	75.56 ± 8.40	78.33	95.83 ± 1.68	94.58	98.40 ± 0.60	98.00	86.35
AdaBoost	78.50 ± 3.74	82.02	88.08 ± 2.66	92.72	84.07 ± 4.48	75.00	78.89 ± 3.63	78.33	96.94 ± 1.12	96.25	99.56 ± 0.28	99.60	87.32
GBDT	78.75 ± 4.03	76.40	86.87 ± 2.12	90.00	80.00 ± 2.16	78.33	73.70 ± 6.02	81.66	96.30 ± 1.17	95.83	98.84 ± 0.60	98.40	86.77
XGBoost	76.75 ± 3.59	75.28	87.27 ± 2.18	89.09	78.89 ± 4.32	73.33	74.07 ± 4.83	78.33	95.19 ± 1.59	95.83	99.11 ± 0.63	99.20	85.18
LightGBM	76.25 ± 3.79	84.26	86.87 ± 2.56	90.00	79.26 ± 2.16	78.33	75.93 ± 4.83	76.66	96.02 ± 0.86	96.66	99.02 ± 0.33	98.40	87.39
Stacking	84.25 ± 3.41	84.26	87.47 ± 0.81	91.81	82.59 ± 4.48	75.00	75.56 ± 3.95	73.33	98.06 ± 0.68	97.91	99.64 ± 0.33	100.00	87.05
Algorithm	**Sample Entropy**												
SVM	81.25 ± 5.70	71.91	87.27 ± 4.81	92.72	79.26 ± 2.72	70.00	88.03 ± 2.59	90.76	95.93 ± 1.26	97.91	99.11 ± 0.40	99.20	87.08
KNN	78.00 ± 4.37	73.03	85.25 ± 2.60	93.63	80.37 ± 2.77	76.66	85.98 ± 3.69	91.53	95.19 ± 1.57	97.50	99.20 ± 0.76	98.40	88.46
RF	76.25 ± 3.16	74.15	84.85 ± 3.26	89.09	76.30 ± 4.12	75.00	84.10 ± 2.84	84.61	95.37 ± 1.21	95.00	99.38 ± 0.45	98.00	85.98
Voting	80.75 ± 4.08	74.15	88.28 ± 3.04	95.45	80.37 ± 2.77	73.33	85.81 ± 2.51	90.00	96.67 ± 1.42	98.75	99.73 ± 0.22	99.60	88.55
Bagging	77.00 ± 3.67	71.91	82.42 ± 3.48	89.09	76.67 ± 0.91	71.66	83.08 ± 3.48	83.07	95.37 ± 0.88	95.83	99.20 ± 0.52	97.20	84.79
AdaBoost	82.25 ± 3.48	77.50	86.26 ± 3.53	91.81	80.37 ± 3.43	71.66	85.30 ± 1.74	87.69	96.57 ± 1.19	97.91	99.64 ± 0.33	98.80	87.56
GBDT	75.00 ± 3.35	77.52	85.25 ± 1.87	87.27	75.93 ± 5.62	70.00	81.54 ± 2.54	83.07	96.20 ± 1.35	97.08	99.02 ± 0.52	97.60	85.42
XGBoost	77.75 ± 4.43	74.15	82.83 ± 4.91	88.18	81.11 ± 2.96	65.00	80.85 ± 2.45	80.00	94.26 ± 0.81	95.00	99.11 ± 0.56	97.60	83.32
LightGBM	77.50 ± 2.74	74.15	81.21 ± 4.93	86.36	78.52 ± 5.05	68.33	78.97 ± 3.32	80.00	94.35 ± 0.80	95.41	98.58 ± 0.59	97.60	83.64
Stacking	79.00 ± 5.78	74.15	88.89 ± 4.09	95.45	81.11 ± 2.46	71.66	88.03 ± 1.95	90.76	96.39 ± 1.26	98.33	99.91 ± 0.18	99.20	88.26
Algorithm	**Permutation Entropy**												
SVM	74.75 ± 5.15	71.91	84.04 ± 2.81	90.90	82.42 ± 3.81	86.36	82.74 ± 2.12	82.30	95.56 ± 0.86	95.41	97.60 ± 0.87	98.80	87.61
KNN	72.50 ± 5.00	66.29	82.22 ± 3.23	90.00	84.04 ± 3.46	88.18	80.85 ± 2.68	86.15	94.35 ± 1.15	95.83	97.24 ± 0.65	97.20	87.28
RF	77.25 ± 2.67	79.77	81.41 ± 2.18	88.18	83.03 ± 1.85	83.63	74.53 ± 3.17	83.84	94.81 ± 1.74	96.25	97.69 ± 1.21	99.20	88.48
Voting	76.75 ± 3.02	69.66	82.83 ± 2.47	90.00	85.05 ± 3.75	90.90	81.37 ± 2.93	86.92	95.56 ± 0.95	97.08	79.02 ± 2.10	49.60	80.69
Bagging	77.50 ± 2.62	79.77	83.43 ± 2.68	89.09	83.03 ± 2.96	83.63	74.53 ± 1.98	86.15	93.33 ± 1.65	93.33	97.87 ± 0.71	98.80	88.46
AdaBoost	77.25 ± 2.15	75.28	84.65 ± 2.16	90.90	84.04 ± 4.30	88.18	77.09 ± 1.47	87.69	95.28 ± 1.48	96.25	99.11 ± 0.71	98.80	89.52
GBDT	77.50 ± 2.85	79.77	81.21 ± 2.90	86.36	83.03 ± 4.67	82.72	74.70 ± 2.89	86.92	93.70 ± 2.14	93.33	98.58 ± 0.86	98.80	87.98
XGBoost	77.00 ± 3.41	79.77	81.01 ± 1.34	87.27	83.84 ± 2.47	81.81	76.24 ± 4.37	86.15	93.80 ± 2.00	94.16	98.58 ± 1.30	98.00	87.86
LightGBM	77.00 ± 2.57	74.15	80.00 ± 2.81	84.54	73.94 ± 5.24	75.45	73.50 ± 2.59	88.46	93.52 ± 2.03	95.00	98.58 ± 1.36	98.40	86.00
Stacking	73.75 ± 5.24	69.66	83.84 ± 2.30	90.90	85.05 ± 3.22	87.27	82.91 ± 3.42	86.15	95.83 ± 0.65	97.08	79.91 ± 2.29	50.40	80.24

respectively. In the case of the F-measure, AdaBoost gained 91.52% for HFD and voting achieved 93.24% for SE features. RF algorithm achieved an average of 89.56% F-measure for the PE feature. In addition, RF, voting, and AdaBoost achieved 89.30%, 92.96%, and 94.47% AUC-ROC for PE, SE, and HFD, respectively. The comparison of average precision, recall, F-measure, and AUC-ROC of each classifier are shown in Fig. 11.

ii) Electrode reduction

We observed an accuracy after reducing the number of electrodes using the RFE method. We observed an accuracy after reducing the number of electrodes using the RFE method. Each classifier's result after electrode reduction is described in Table 10 and Table 11. Random Forest achieved the highest average accuracy of 87.50% for HFD. The voting and AdaBoost achieved 88.55% and 89.52% accuracy for SE and PE, respectively. In addition, The SVM classifier achieved an average of

Table 11
Precision, recall, F-measure and AUC-ROC (%) using AMIGOS dataset with RFE.

Algorithm	Higuchi Fractal Dimension																							
	Sad				Fear				Shamed				Hope				Interest				Excited			
	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR	P	R	F	AR
SVM	71.11	78.04	74.41	75.48	86.88	86.88	86.88	85.27	73.52	86.20	79.36	78.58	96.61	71.25	82.01	83.62	99.17	96.00	97.56	97.56	100	98.41	99.19	99.20
KNN	71.05	65.85	68.35	71.46	87.88	95.08	91.33	89.37	71.42	86.20	78.12	76.97	89.28	93.75	91.46	87.87	95.16	94.40	94.77	94.59	97.63	98.41	98.02	97.99
RF	73.33	80.48	76.74	77.74	85.43	86.88	86.17	84.25	70.00	96.55	81.11	78.92	90.90	87.50	89.17	86.75	97.58	96.80	97.18	97.09	98.43	100	99.21	99.19
Voting	75.00	80.48	77.64	78.78	90.32	91.80	91.05	89.77	71.42	86.20	78.12	76.97	91.25	91.25	91.25	88.62	98.37	96.80	97.58	97.53	99.21	100	99.60	99.59
Bagging	73.80	75.60	74.69	76.34	85.07	93.44	89.06	86.51	71.79	96.55	82.35	80.53	88.31	85.00	86.62	83.50	96.74	95.20	95.69	95.86	96.18	100	98.05	97.98
AdaBoost	75.00	80.48	77.64	78.78	91.93	93.44	92.68	91.61	74.28	89.65	81.25	80.31	91.25	91.25	91.25	88.62	96.00	96.00	96.00	95.82	98.43	100	99.21	991.9
GBDT	71.42	73.17	72.28	74.08	86.88	86.88	86.88	85.27	68.42	89.65	77.61	75.47	92.10	87.50	89.74	87.75	97.61	98.40	98.00	97.89	98.43	100	99.21	991.9
XGBoost	76.31	70.73	73.41	75.99	89.83	86.88	88.33	87.32	71.42	86.20	78.12	76.97	84.41	81.25	82.80	78.62	95.23	96.00	95.61	95.39	96.15	99.20	97.65	97.58
LightGBM	71.05	65.85	68.35	71.46	87.09	88.52	87.80	86.09	68.57	82.75	75.00	73.63	90.14	80.00	84.76	83.00	97.58	96.80	97.18	97.09	96.92	100	98.43	98.38
Stacking	70.45	75.60	72.94	74.26	91.66	90.16	90.90	89.97	69.44	86.20	76.92	75.36	94.80	91.25	92.99	91.62	98.37	96.80	97.58	97.53	100	98.41	99.19	99.20
Algorithm	**Sample Entropy**																							
SVM	73.80	75.60	74.98	76.34	95.00	93.44	94.21	93.66	68.75	75.86	72.13	71.80	93.82	95.00	94.40	92.49	95.27	96.80	96.03	95.79	100	98.41	99.19	99.20
KNN	75.60	75.60	75.60	77.38	91.52	88.52	90.00	89.16	74.28	89.65	81.25	80.13	93.42	88.75	91.02	89.37	95.27	96.80	96.03	95.79	97.63	98.41	98.02	97.99
RF	76.92	73.17	74.99	77.21	90.00	88.52	89.25	88.13	67.64	79.31	73.01	71.91	97.18	86.25	91.39	91.12	97.54	95.20	96.35	96.29	97.63	98.41	98.02	97.99
Voting	76.19	78.04	77.10	78.60	74.13	70.49	72.26	69.93	73.52	86.20	79.36	78.58	93.33	87.50	90.32	88.75	96.00	96.00	96.00	96.00	100	98.41	99.19	99.20
Bagging	81.08	73.17	76.92	79.29	96.42	88.52	92.30	92.22	69.69	79.31	74.19	73.52	90.54	83.75	87.01	84.87	98.33	94.40	96.32	96.33	96.15	99.20	97.65	97.58
AdaBoost	76.19	78.04	77.10	78.60	87.50	91.80	89.59	87.73	68.57	82.75	75.00	73.63	92.30	90.00	91.13	89.00	98.34	95.20	96.74	96.73	99.20	99.20	99.20	99.19
GBDT	83.78	75.60	79.48	81.55	88.70	90.16	89.43	87.93	63.63	72.41	67.74	66.85	90.41	82.50	86.27	84.24	97.52	94.40	95.93	95.89	96.89	99.20	98.03	97.99
XGBoost	75.00	73.17	74.07	76.16	83.58	91.80	87.50	84.67	71.87	79.31	75.40	75.13	89.33	83.75	86.45	83.87	95.16	94.40	94.77	94.59	96.89	99.20	98.03	97.99
LightGBM	75.00	73.17	74.07	76.16	88.88	91.80	90.32	88.75	67.64	79.31	73.00	71.91	91.89	85.00	88.31	86.50	95.12	93.60	94.35	94.19	96.89	99.20	98.03	97.99
Stacking	76.31	70.73	73.41	75.99	96.36	86.88	91.37	91.40	72.72	82.75	77.41	76.86	93.42	88.75	91.02	89.37	98.36	96.00	97.16	97.13	100	98.41	99.19	99.20
Algorithm	**Permutation Entropy**																							
SVM	65.78	60.97	63.29	66.94	96.49	90.16	93.22	93.04	65.78	86.20	74.62	72.13	93.84	76.25	84.13	84.12	98.33	94.40	96.32	96.33	100	93.65	96.72	96.82
KNN	61.36	65.85	63.52	65.21	93.33	91.80	92.56	91.82	66.66	82.75	73.84	72.02	87.20	93.75	90.36	85.87	93.23	99.20	96.12	95.68	95.38	98.41	96.87	96.78
RF	76.19	78.04	77.10	78.60	91.80	91.80	91.80	90.79	63.33	65.51	64.40	65.01	89.87	88.75	89.30	86.37	95.93	94.40	95.16	95.02	99.18	96.82	97.99	98.00
Voting	65.95	75.60	70.45	71.13	93.33	91.80	92.56	91.82	65.00	89.65	75.36	72.24	89.02	91.25	90.12	86.62	97.60	97.60	97.60	97.60	99.20	98.41	98.80	98.80
Bagging	79.48	75.60	77.49	79.47	91.93	93.44	92.68	91.61	69.69	68.95	67.79	68.35	87.80	90.00	88.88	85.00	90.76	94.40	92.54	91.98	99.18	96.82	97.99	98.00
AdaBoost	68.18	73.17	70.58	72.00	94.82	90.16	92.43	92.02	68.75	75.86	72.13	71.80	91.02	88.75	89.87	87.37	96.80	96.80	96.80	96.80	98.41	98.41	98.41	98.39
GBDT	76.19	78.04	77.10	78.60	91.52	88.52	90.00	89.16	66.66	62.06	64.28	66.51	89.18	82.50	85.71	83.25	95.90	93.60	94.73	94.62	99.20	98.41	98.80	98.80
XGBoost	65.90	70.73	68.23	69.74	88.33	86.88	87.60	86.29	67.64	79.31	73.01	71.91	89.70	76.25	82.43	81.12	94.35	93.60	93.97	93.75	98.40	97.61	98.00	98.00
LightGBM	75.60	75.60	75.60	77.38	88.33	86.88	87.60	86.29	65.51	65.51	65.51	66.62	88.73	78.75	83.44	81.37	93.54	92.80	93.17	92.92	99.19	97.61	98.40	98.40
Stacking	61.90	63.41	62.65	65.04	98.18	88.52	93.10	93.24	70.96	75.86	73.33	73.41	86.41	87.50	86.95	82.75	97.54	95.20	96.35	96.29	99.18	96.03	97.58	97.61

Classification performance metrics: Precision (P), Recall (R), F-Measure (F), AUC-ROC (AR).

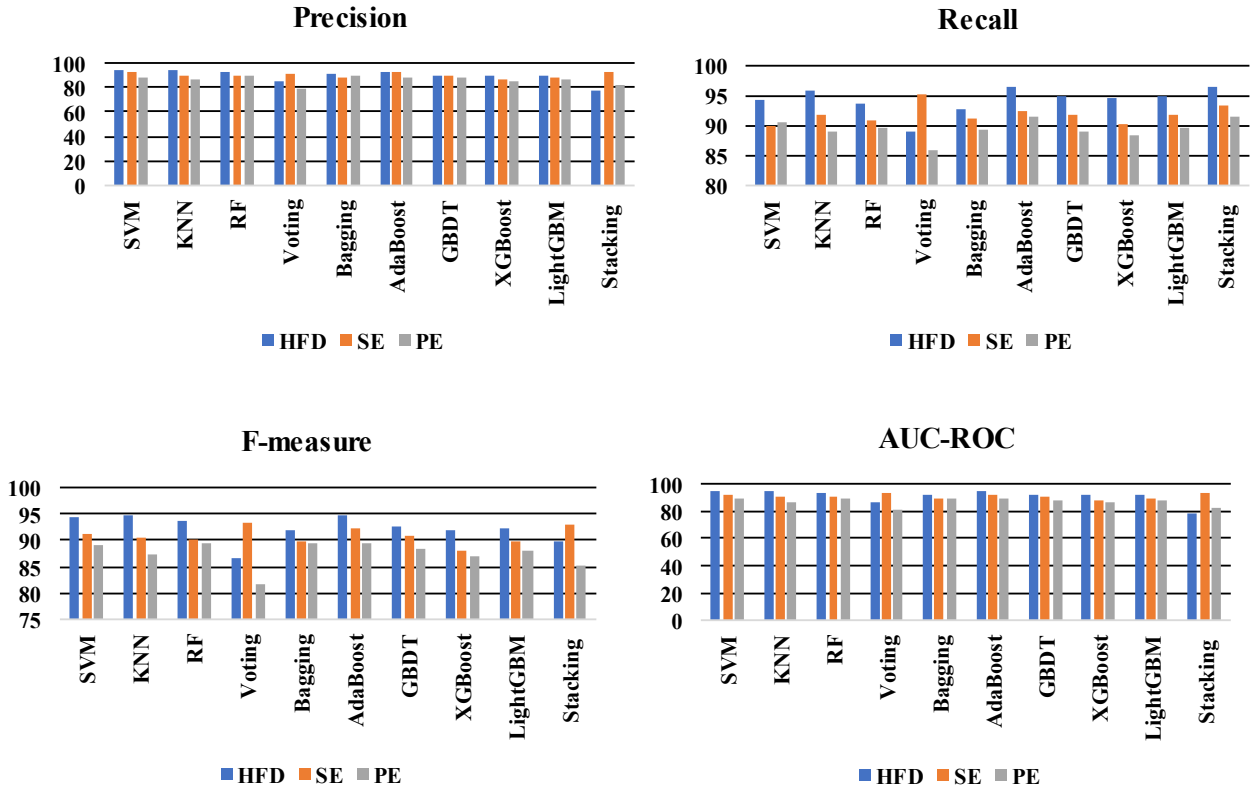


Fig. 12. Average precision, recall, F-measure and AUC-ROC for each classifier after electrode reduction.

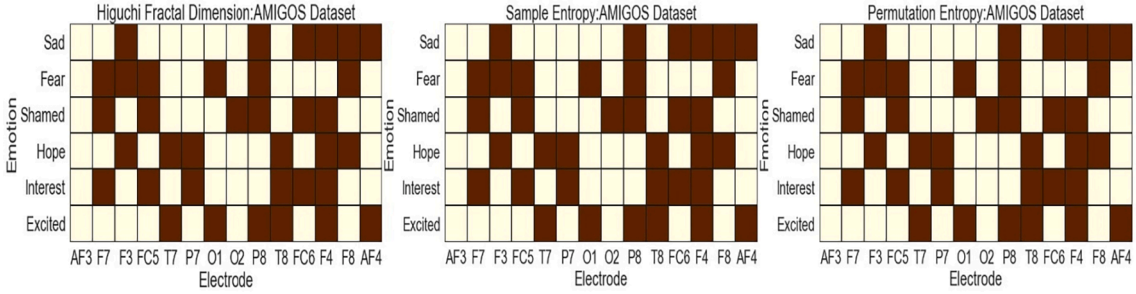
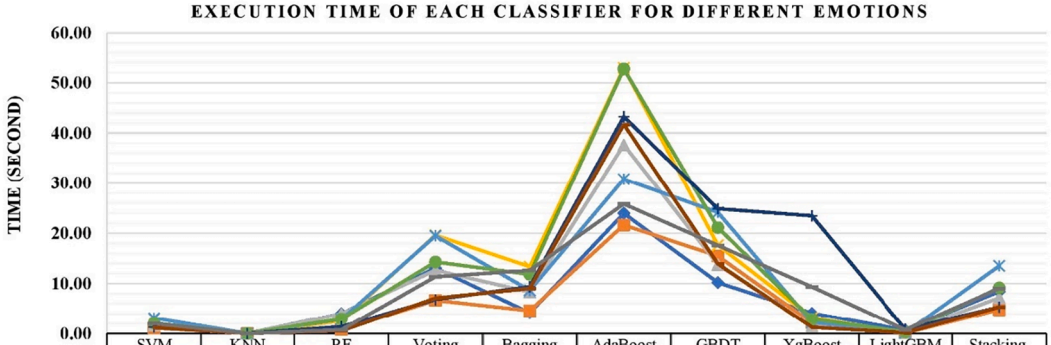


Fig. 13. Heat map of the selected channels for each emotion using three non-linear features on AMIGOS dataset. The red color in the square box indicates the presence of electrodes for each emotion.

94.16% and 92.54% precision for HFD and SE, respectively. The RF achieved an average of 89.51% precision for PE. The voting classifier obtained an average of 95.17% recall for the SE feature. The stacking classifier achieved an average of 94.46% and 91.46% recall for HFD, and PE, respectively. Regarding the F-measure, the RF classifier performed well using PE and which was 89.56%. In addition, the voting and AdaBoost achieved 93.24% and 94.65% for SE and HFD, respectively. Similarly, in the case of AUC-ROC, the RF algorithm gained 89.30% using PE, whereas the voting and AdaBoost classifiers obtained an average of 92.96% and 94.47% for the SE and HFD features. The comparison of average precision, recall, F-measure, and AUC-ROC of each classifier are shown in Fig. 12. Furthermore, it was observed that the average accuracy considering all electrodes was higher than that of the accuracy obtained using a reduced number of electrodes. Therefore, we can conclude that the use of more electrodes is superior to a smaller number of electrodes to predict emotional states. The heat map of electrodes related to each emotion using different features is illustrated in Fig. 13.

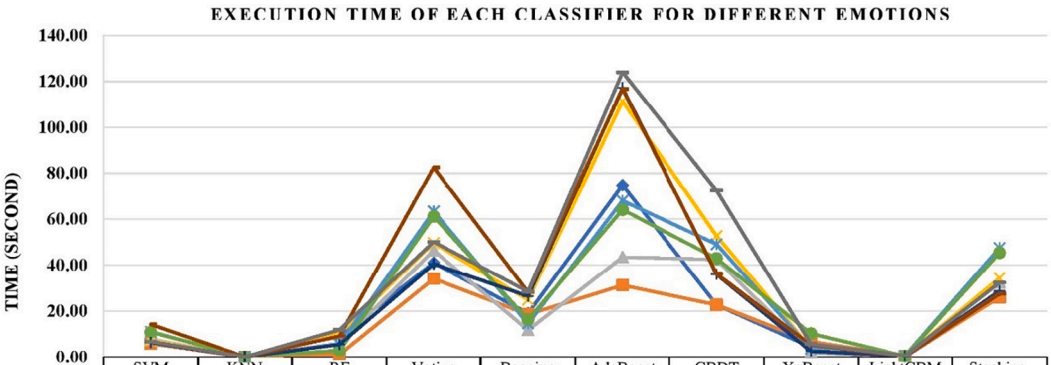
c. Execution time

Traditionally, ensemble methods are less complex than deep learning methods. Consequently, in this paper, we focused on the execution time of each ensemble method for performing each experiment on the DEAP dataset. The execution time of each classifier on test data is shown in Fig. 14 and Fig. 15. In both figures, it was found that the execution time of the SVM classifier on test data is less than that of the other ensemble methods except the LightGMB classifier. Although KNN took less time to execute each experiment, it did not achieve the highest classification accuracy as other classifiers. Besides, LightGBM showed better classification accuracy with less execution time. According to Fig. 14, for sad, the execution time of LightGBM is 0.65 s using HFD, which is 92.30% lower than the execution time of the stacking method. In contrast, the accuracy of LightGBM is 87.50%, which is 4.53% lower than the accuracy of the stacking classifier, and the same things happened in all the experiments shown in Fig. 14 and Fig. 15. Therefore, in the case of execution time, LightGMB could be the efficient classifier, whereas, in the case of accuracy, a stacking classifier could be the best choice for



	SVM	KNN	RF	Voting	Bagging	AdaBoost	GBDT	XgBoost	LightGBM	Stacking
Sad Higuchi	1.78	0.02	3.98	13.11	4.14	23.97	10.19	4.07	0.65	8.42
Sad Sample	1.09	0.02	0.79	6.68	4.57	21.65	15.49	2.17	0.08	4.81
Sad Permutation	1.65	0.02	3.95	12.69	8.31	37.66	13.76	1.53	0.13	7.16
Fear Higuchi	3.07	0.02	2.49	19.63	13.37	52.94	17.64	3.16	0.15	13.51
Fear Sample	3.09	0.02	2.86	19.50	8.64	30.82	24.15	2.25	0.16	13.52
Fear Permutation	2.08	0.02	2.96	14.32	11.84	52.76	21.13	2.89	0.31	9.16
Shamed Higuchi	1.31	0.02	1.36	6.84	9.40	43.27	24.90	23.47	0.84	5.32
Shamed Sample	1.20	0.02	0.54	7.11	9.05	41.72	13.91	1.25	0.16	5.34
Shamed Permutation	1.98	0.02	0.74	11.37	12.61	25.84	17.61	9.35	0.76	9.05

Fig. 14. Execution time of each classifier (second) for each of the emotions using thirty-two electrodes.



	SVM	KNN	RF	Voting	Bagging	AdaBoost	GBDT	XgBoost	LightGBM	Stacking
Hope Higuchi	5.90	0.02	9.20	40.95	19.01	74.78	22.67	4.32	0.19	26.23
Hope Sample	5.83	0.02	1.10	34.05	18.51	31.26	22.66	6.67	0.37	26.00
Hope Permutation	7.69	0.05	6.05	46.34	11.74	43.44	42.43	1.82	0.31	32.36
Interest Higuchi	7.19	0.05	10.20	49.63	24.91	111.39	52.62	3.80	0.36	34.30
Interest Sample	10.93	0.06	5.58	63.29	14.51	68.01	49.04	3.59	0.49	47.40
Interest Permutation	10.81	0.06	2.82	61.12	16.20	64.10	42.83	10.19	0.36	45.37
Excited Higuchi	6.29	0.02	5.41	40.51	26.50	117.10	35.96	2.55	0.45	28.62
Excited Sample	14.13	0.10	9.07	82.57	28.14	116.72	36.12	5.14	0.30	27.50
Excited Permutation	6.70	0.03	11.83	50.03	28.50	123.96	72.43	5.66	0.57	32.31

Fig. 15. Execution time of each classifier (second) for each of the emotions after reducing the number of electrodes.

classifying the emotions.

d. Comparative analysis

After analyzing the results, it is seen that AMIGOS outperformed the DEAP dataset using the same methods in some cases. Fig. 16 represents the comparative analysis of average accuracy on both datasets using all classifiers and features. The above figure shows that, for sample entropy using all electrodes, the average accuracy of all classifiers is higher on the AMIGOS dataset than the DEAP dataset. Similarly, according to Fig. 17, for three features using optimum electrodes, some classifiers' average accuracy is higher on the AMIGOS dataset than the DEAP dataset.

A comparison of our research work with previous works is shown in Table 12. Most studies have used EEG signals to identify discrete/ basic emotions based on valence and arousal dimensions, and some of the studies used multimodal signals. M. Murugappan et al. (Murugappan et al., 2011) used KNN and LDA for identifying five emotions using spatial filtering, wavelet, and time–frequency features. The average accuracy was 83.04% for 64 electrodes and 79.17% for 24 electrodes which were 6.34% and 9.45% lower than our proposed method. Rules algorithm, DT, and SVM were applied with HFD and PSD for four emotional states, and an average accuracy of 65.66% was achieved for all the experiments on the DEAP dataset (Pane et al., 2018). Y. J. Liu et al. (Liu et al., 2018) classified a total of seven emotions; among them, four were belonged to negative emotions, and three were positive emotions. Time-frequency (TF) features were implemented and an accuracy of 65.09% for negative emotions and 86.43% for positive emotions were obtained. The overall accuracy for both emotions was 75.76% which was 13.62% lower than our proposed method.

Similarly, many of the studies worked out with deep learning methods and achieved better accuracy. Hybrid Long Short-Term Memory (LSTM) algorithm was used with two (wavelet features, time-domain features) features for classifying emotions where emotions

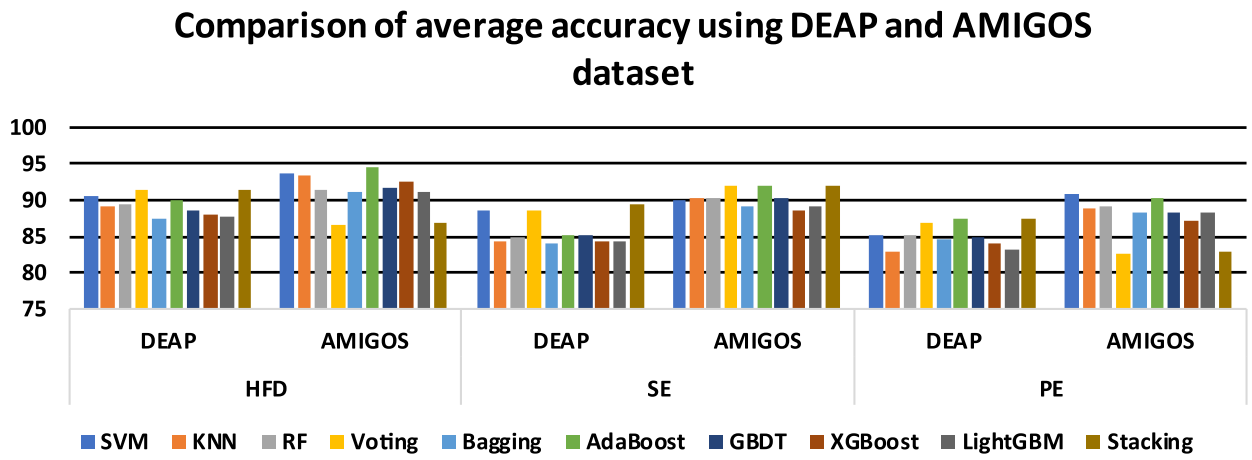


Fig. 16. Comparative analysis of average accuracy by considering all electrodes on both datasets.

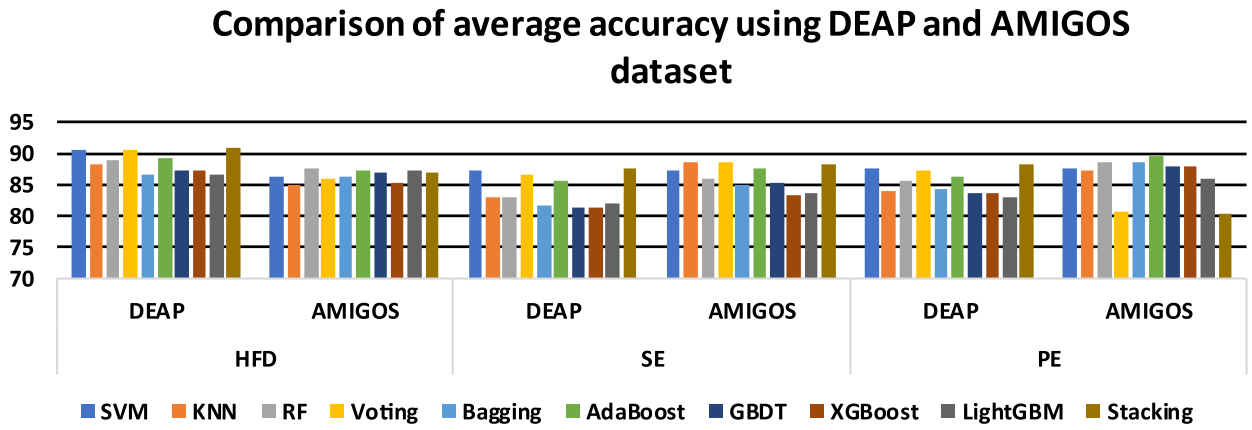


Fig. 17. Comparative analysis of average accuracy by considering optimum electrodes on both datasets.

were stimulated by using 3-D VR videos (Thejaswini & Ravikumar, 2020). The average accuracy was 80.05% which is 9.33% lower than our proposed method when the same number of electrodes (32) were used in both studies. Moreover, S. Gannouni et al. (Gannouni et al., 2021) worked on the DEAP dataset for eight emotions and achieved an average accuracy of 87.63% after applying zero-time windowing (ZTW). It was observed that our proposed method achieved 1.75% more accuracy than that of studies (Gannouni et al., 2021). In contrast, articles (Lu et al., 2021) and (Zheng et al., 2019) applied deep learning methods for detecting six and four emotions, respectively, where both studies were used multimodal features, among them entropy analysis, eye movement, and facial expression features were important. However, we achieved 0.38% and 18.29% more accuracy than studies (Lu et al., 2021) and (Zheng et al., 2019), respectively. As a result, our proposed method ensures the possibility of classifying discrete emotions by using non-linear features and ensemble methods. Moreover, it can be consolidated that, the combination of non-linear features and the RFE method was a good choice for classifying emotions after tunning the hyper-parameter of each of the algorithms.

e. Statistical test of significance

We have used two statistical methods named Combined 5 × 2 cv F test and Friedman test to test the statistical significance and confirm that the outcomes were accurate, dependable, and did not happen by coincidence. The first one compares the algorithms to each other, and the second one compares the algorithms over two datasets. In both cases, we calculated the p-values of all algorithms based on accuracy using all and optimum electrodes. Table 13 depicts the outcome of the p-values for comparison between two algorithms. The p-value was calculated based on accuracy on the AMIGOS dataset while analyzing negative emotion (sad). In this case, the accuracy of each algorithm has been calculated using permutation entropy. From Table 13, the p-value among voting and two other methods (XGBoost and LightGBM) is lower than 0.005. Therefore, we can reject the null hypothesis and say that voting and XGBoost are not identical. Similarly, voting and LightGBM have a significant difference.

Furthermore, we performed the Friedman test to compare all the algorithms over two datasets. In this test, it was observed that the performance of all the algorithms is almost identical, which means our proposed methods performed identically on both datasets, and the p-values for pairwise comparison using all features are depicted in Table 14.

6. Conclusion

This paper proposed ensemble methods for detecting six (hope, interest, excitement, shame, fear, and sad) emotions using three non-linear features. The proposed approaches showed better overall accuracy than other techniques dealing with shallow machine learning techniques.

Table 12
A comparative study with related works based on average accuracy.

Reference	Dataset/ Stimuli	Number of electrodes	Extracted Feature	Emotion		Classifier	Average accuracy (%)
				No's	**Type**		
(Murugappan et al., 2011)	Audio-Visual	64 24	Spatial Filtering, Wavelet, Time-Frequency Analysis	5	Happy, Disgust, Surprise, Fear, Neutral	KNN, LDA	83.04 79.17
(Pane et al., 2018)	DEAP	5	Higuchi fractal dimension and Power spectral density (PSD) using Welch's method	4	Happy, sad, angry, relaxed	Rules algorithm (RIPPER)DT (J4.8) SVM	65.66
(Liu et al., 2018)	Audio-Visual	14	Time-Frequency (TF) Features	7	Anger, disgust, fear, sadness, Joy, amusement, tenderness	Real time system	65.09 86.43
(Thejaswini & Ravikumar, 2020)	3-D VR videos	32	Wavelet features, Time-domain features	8	Happy, excited, calm, bored, fear, tensed, sad, relax	Hybrid Long Short-Term Memory (LSTM) algorithm	80.05
(Ahirwal & Kose, 2018)	DEAP	32	Time domain features Frequency domain features Entropy	4	Happy, sad, angry, and relaxed	SVM, ANN, NB	81.16
(Gannouni et al., 2021)	DEAP	Adaptive	ZTWBES	8	Happy, Pleased, relaxed, excited, calm, distressed, miserable, depressed	QDC RNN-scheme 1 RNN-scheme 2	87.63 89.33 86.53
(Lu et al., 2021)	MAHNOB-HCI	32	Facial expression features, Spectral energy	6	Anger, disgust, fear, happiness, sadness, and surprise	VGG-LSTM network	89
(Zheng et al., 2019)	Audio-Visual	6	PSD and DE for EEG signal, Statistical feature for eye movement	4	Happy, sad, fear, and neutral	EmotionMeter with DNN	85.11 (Both) 70.33 (EEG) 67.82 (Eye movement)
Proposed Method	DEAP AMIGOS	32 Adaptive 14 Adaptive	Higuchi fractal dimension (HFD)Sample entropy (SE) Permutation entropy (PE)	6	Hope, interest, excited, shamed, fear, sad	SVM, KNN, RF, Voting, Bagging, AdaBoost, GBDT, XGBoost, LightGBM, Stacking	89.38 (HFD) 88.62 (HFD) 94.62 (HFD) 87.50 (HFD)

Table 13
Combined 5 × 2 cv F test results.

	SVM	KNN	RF	Voting	Bagging	AdaBoost	GBDT	XGBoost	LightGBM	Stacking
SVM										
KNN	0.783									
RF	0.565	0.625								
Voting	0.307	0.611	0.229							
Bagging	0.525	0.679	0.468	0.189						
AdaBoost	0.728	0.747	0.642	0.155	0.608					
GBDT	0.587	0.771	0.392	0.474	0.096	0.486				
XGBoost	0.636	0.803	0.453	0.030	0.266	0.404	0.305			
LightGBM	0.384	0.067	0.069	0.004	0.032	0.195	0.106	0.219		
Stacking	0.640	0.508	0.607	0.413	0.624	0.734	0.672	0.752	0.408	

Table 14
Friedman test results on both datasets for average accuracy of each classifier.

	Without RFE			With RFE		
Dataset (pairwise)	DEAP-AMIGOS	DEAP-AMIGOS	DEAP-AMIGOS	DEAP-AMIGOS	DEAP-AMIGOS	DEAP-AMIGOS
Feature	HFD	SE	PE	HFD	SE	PE
p-value	0.432	0.088	0.412	0.432	0.073	0.604

Eight ensemble approaches were used to classify emotions by adjusting the hyperparameters of each of the classifiers. Moreover, in this paper, we have used the recursive feature elimination method to select the relevant electrodes to each emotion. As a result, the recursive feature elimination method significantly reduced the number of electrodes and provided constant accuracy. Amongst those, the stacking ensemble method showed the highest overall average accuracy. The average overall accuracy was 89.38% when thirty-two electrodes were used, whereas we achieved an overall accuracy of 88.62% using adaptive electrodes. Compared to other current methods, the proposed method obtained better accuracy with and without feature selection technique. In addition, we have also tested the model's performance using the AMIGOS dataset. We achieved an average accuracy of 94.62% and 87.50% for all electrodes and optimal electrodes. Moreover, precision, recall, F-measure, and AUC-ROC of different algorithms have been calculated. The results of each classifier's performance metrics have been described in the graph and table.

In the future, we can use multimodal approaches to develop this research and increase the accuracy. Moreover, we intend to use multi-class classification to identify the emotion. Furthermore, future research will focus on making new datasets like DEAP and AMIGOS to analyze emotional states.

CRediT authorship contribution statement

Md. Mustafizur Rahman: Conceptualization, Methodology, Software, Writing – original draft, Data curation, Writing – review & editing. **Ajay Krishno Sarkar:** Resources, Formal analysis, Project administration. **Md. Amzad Hossain:** Validation, Resources, Investigation. **Mohammad Ali Moni:** Visualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ahirwal, M. K., & Kose, M. R. (2018). Emotion recognition system based on EEG signal: A comparative study of different features and classifiers. In *Proceedings of the 2nd International Conference on Computing Methodologies and Communication*. https://doi.org/10.1109/ICCMC.2018.8488044

Alotaiby, T., El-Samie, F. E. A., Alshebeili, S. A., & Ahmad, I. (2015). A review of channel selection algorithms for EEG signal processing. *Eurasip Journal on Advances in Signal Processing, 2015*(1). https://doi.org/10.1186/s13634-015-0251-9

Alpaydin, E. (1999). Combined 5 × 2 cv F test for comparing supervised classification learning algorithms. *Neural Computation, 11*(8), 1885–1892. https://doi.org/10.1162/089976699300016007

Altman, N. S. (1992). An introduction to Kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175. https://doi.org/10.2307/2685209

Bălan, O., Moise, G., Moldoveanu, A., Leordeanu, M., & Moldoveanu, F. (2019). Fear Level Classification Based on Emotional Dimensions and Machine Learning Techniques. *Sensors 2019, 19, Page 1738*, *19*(7), 1738. https://doi.org/10.3390/S19071738.

Bălan, O., Moise, G., Petrescu, L., Moldoveanu, A., Leordeanu, M., & Moldoveanu, F. (2020). Emotion classification based on biophysical signals and machine learning techniques. *Symmetry, 12*(1), 1–22. https://doi.org/10.3390/sym12010021

Bandt, C., & Pompe, B. (2002). Permutation entropy: A natural complexity measure for time series. *Physical Review Letters, 88*(17), Article 174102. https://doi.org/10.1103/PhysRevLett.88.174102

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter. *Optimization., 13*, 281–305.

Breiman, L. (1996). Bagging predictors. *Machine Learning 1996 24:2*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655.

Breiman, L. (2001). Random Forests. *Machine Learning 2001 45:1*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/JAIR.953

Chen, G., Zhang, X., Sun, Y., & Zhang, J. (2020). Emotion feature analysis and recognition based on reconstructed EEG sources. *IEEE Access, 8*, 11907–11916. https://doi.org/10.1109/ACCESS.2020.2966144

Chen, J. X., Zhang, P. W., Mao, Z. J., Huang, Y. F., Jiang, D. M., & Zhang, Y. N. (2019). Accurate EEG-based emotion recognition on combined features using deep convolutional neural networks. *IEEE Access, 7*, 44317–44328. https://doi.org/10.1109/ACCESS.2019.2908285

Chen, Q., Meng, Z., Liu, X., Jin, Q., & Su, R. (2018). Decision variants for the automatic determination of optimal feature subset in RF-RFE. *Genes, 9*(6). https://doi.org/10.3390/genes9060301

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Dogan, A. (n.d.). A Weighted Majority Voting Ensemble Approach for Classification. *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 1–6. https://doi.org/10.1109/UBMK.2019.8907028.

Ekman, P. (1972). Universals and Cultural Differences in Facial Expressions of Emotion BT - Nebraska Symposium on Motivation. In *Nebraska Symposium on Motivation* (Vol. 19, pp. 207–282). papers3://publication/uuid/FDC5E29A-0E28-4DDF-B1A4-F53FEE0B4F70.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139. https://doi.org/10.1006/JCSS.1997.1504

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Gannouni, S., Aledaily, A., Belwafi, K., & Aboalsamh, H. (2021). Emotion detection using electroencephalography signals and a zero-time windowing-based epoch estimation and relevant electrode identification. *Scientific Reports, 11*(1), 1–17. https://doi.org/10.1038/s41598-021-86345-5

Gao, Z., Cui, X., Wan, W., & Gu, Z. (2019). Recognition of emotional states using multiscale information analysis of high frequency EEG oscillations. *Entropy 2019, Vol. 21, Page 609*, *21*(6), 609. https://doi.org/10.3390/E21060609.

García-Martínez, B., Martínez-Rodrigo, A., Cantabrana, R. Z., García, J. M. P., & Alcaraz, R. (2016). Application of entropy-based metrics to identify emotional distress from electroencephalographic recordings. *Entropy 2016, Vol. 18, Page 221*, *18*(6), 221. https://doi.org/10.3390/E18060221.

Higuchi, T. (1988). Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena, 31*(2), 277–283. https://doi.org/10.1016/0167-2789(88)90081-4

Hossain, M. S. (2017). Cloud-supported cyber-physical localization framework for patients monitoring. *IEEE Systems Journal, 11*(1), 118–127. https://doi.org/10.1109/JSYST.2015.2470644

Huang, D., Guan, C., Ang, K. K., Zhang, H., & Pan, Y. (2012). Asymmetric Spatial Pattern for EEG-based emotion detection. In *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2012.6252390

Huang, J.-R., Fan, S.-Z., Abbod, M. F., Jen, K.-K., Wu, J.-F., & Shieh, J.-S. (2013). Application of Multivariate Empirical Mode Decomposition and Sample Entropy in EEG Signals via Artificial Neural Networks for Interpreting Depth of Anesthesia. *Entropy 2013, Vol. 15, Pages 3325-3339*, *15*(9), 3325–3339. https://doi.org/10.3390/E15093325.

Zhang, J., Chen, M., Zhao, S., Hu, S., Shi, Z., & Cao, Y. (2016). ReliefF-based EEG sensor selection methods for emotion recognition. *Sensors (Basel Switzerland), 16*(10). https://doi.org/10.3390/S16101558

Jenke, R., Peer, A., & Buss, M. (2014). Feature extraction and selection for emotion recognition from EEG. *IEEE Transactions on Affective Computing, 5*(3), 327–339. https://doi.org/10.1109/TAFFC.2014.2339834

Jiang, G. J. A., Fan, S. Z., Abbod, M. F., Huang, H. H., Lan, J. Y., Tsai, F. F., … Shieh, J. S. (2015). Sample entropy analysis of EEG signals via artificial neural networks to model patients' consciousness level based on anesthesiologists experience. *BioMed Research International, 2015*. https://doi.org/10.1155/2015/343478

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 3147–3155.

Khare, S. K., & Bajaj, V. (2021). An evolutionary optimized variational mode decomposition for emotion recognition. *IEEE Sensors Journal, 21*(2), 2035–2042. https://doi.org/10.1109/JSEN.2020.3020915

Koelstra, S., Mühl, C., Soleymani, M., Lee, J. S., Yazdani, A., Ebrahimi, T., … Patras, I. (2012). DEAP: A database for emotion analysis; Using physiological signals. *IEEE Transactions on Affective Computing, 3*(1), 18–31. https://doi.org/10.1109/T-AFFC.2011.15

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2007). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review 2007 26:3*, *26*(3), 159–190. https://doi.org/10.1007/S10462-007-9052-3.

Lan, Z., Sourina, O., Wang, L., & Liu, Y. (2015). Real-time EEG-based emotion monitoring using stable features. *The Visual Computer 2015 32:3*, *32*(3), 347–358. https://doi.org/10.1007/S00371-015-1183-Y.

Li, M., & Lu, B. L. (2009). Emotion classification based on gamma-band EEG. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine*. https://doi.org/10.1109/IEMBS.2009.5334139

Liu, Y. J., Yu, M., Zhao, G., Song, J., Ge, Y., & Shi, Y. (2018). Real-time movie-induced discrete emotion recognition from EEG signals. *IEEE Transactions on Affective Computing, 9*(4), 550–562. https://doi.org/10.1109/TAFFC.2017.2660485

Lu, Y., Zhang, H., Shi, L., Yang, F., & Li, J. (2021). Expression-EEG bimodal fusion emotion recognition method based on deep learning. *Computational and Mathematical Methods in Medicine, 2021*. https://doi.org/10.1155/2021/9940148

Mehmood, R. M., Du, R., & Lee, H. J. (2017). Optimal feature selection and deep learning ensembles method for emotion recognition from human brain EEG sensors. *IEEE Access, 5*, 14797–14806. https://doi.org/10.1109/ACCESS.2017.2724555

Mehrabian, A., & Russell, J. A. (1977). Evidence for a three-factor theory of emotions. *Journal of Research in Personality, 11*, 273–294.

Miranda-Correa, J. A., Abadi, M. K., Sebe, N., & Patras, I. (2021). AMIGOS: A dataset for affect, personality and mood research on individuals and groups. *IEEE Transactions on Affective Computing, 12*(2), 479–493. https://doi.org/10.1109/TAFFC.2018.2884461

Mohammadi, Z., Frounchi, J., & Amiri, M. (2016). Wavelet-based emotion recognition system using EEG signal. *Neural Computing and Applications 2016 28:8*, *28*(8), 1985–1990. https://doi.org/10.1007/S00521-015-2149-8.

Murugappan, M., Nagarajan, R., & Yaacob, S. (2011). Combining spatial filtering and wavelet transform for classifying human emotions using EEG Signals. *Journal of Medical and Biological Engineering, 31*(1), 45–51. https://doi.org/10.5405/jmbe.710

Myers, D. G. (2003). *Psychology*.

Pane, E. S., Hendrawan, M. A., Wibawa, A. D., & Purnomo, M. H. (2018). Identifying rules for electroencephalograph (EEG) emotion recognition and classification. In *Proceedings of 2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*. https://doi.org/10.1109/ICICI-BME.2017.8537731

Picard, R. W. (2000). *Affective Computing | The MIT Press*. https://mitpress.mit.edu/books/affective-computing.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of Emotion* (pp. 3–33). Academic Press. https://doi.org/10.1016/b978-0-12-558701-3.50007-7.

Qing, C., Qiao, R., Xu, X., & Cheng, Y. (2019). Interpretable emotion recognition using EEG signals. *IEEE Access, 7*, 94160–94170. https://doi.org/10.1109/ACCESS.2019.2928691

Rahman, M. M., Sarkar, A. K., Hossain, M. A., Hossain, M. S., Islam, M. R., Hossain, M. B., … Moni, M. A. (2021). Recognition of human emotions using EEG signals: A review. *Computers in Biology and Medicine, 136*, Article 104696. https://doi.org/10.1016/J.COMPBIOMED.2021.104696

Rahman, S., Irfan, M., Raza, M., Ghori, K. M., Yaqoob, S., & Awais, M. (2020). Performance Analysis of Boosting Classifiers in Recognizing Activities of Daily Living. *International Journal of Environmental Research and Public Health 2020, Vol. 17, Page 1082*, *17*(3), 1082. https://doi.org/10.3390/IJERPH17031082.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *Https://Doi.Org/10.1152/Ajpheart.2000.278.6.H2039, 278*(6 47-6), 2039–2049. https://doi.org/10.1152/AJPHEART.2000.278.6.H2039

Riedl, M., Müller, A., & Wessel, N. (2013). Practical considerations of permutation entropy. *The European Physical Journal Special Topics 2013 222:2*, *222*(2), 249–262. https://doi.org/10.1140/EPJST/E2013-01862-7.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology, 39*(6), 1161–1178. https://doi.org/10.1037/h0077714

Schölkopf, B. (1998). SVMs - A practical consequence of learning theory. *IEEE Intelligent Systems and Their Applications, 13*(4), 18–21. https://doi.org/10.1109/5254.708428

Taran, S., & Bajaj, V. (2019). Emotion recognition from single-channel EEG signals using a two-stage correlation and instantaneous frequency-based filtering method. *Computer Methods and Programs in Biomedicine, 173*, 157–165. https://doi.org/10.1016/J.CMPB.2019.03.015

Thejaswini, S., & Ravikumar, K. M. (2020). Electroencephalogram based emotion detection using hybrid longshort term memory. *European Journal of Molecular and Clinical Medicine, 7*(8), 2786–2792. https://ejmcm.com/article_4791.html.

Torres, E. P., Torres, E. A., Hernández-Álvarez, M., & Yoo, S. G. (2020). Emotion recognition related to stock trading using machine learning algorithms with feature selection. *IEEE Access, 8*, 199719–199732. https://doi.org/10.1109/ACCESS.2020.3035539

Vijayan, A. E., Sen, D., & Sudheer, A. P. (2015). EEG-based emotion recognition using statistical measures and auto-regressive modeling. *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, 587–591. https://doi.org/10.1109/CICT.2015.24.

Wang, X. W., Nie, D., & Lu, B. L. (2014). Emotional state classification from EEG data using machine learning approach. *Neurocomputing, 129*, 94–106. https://doi.org/10.1016/J.NEUCOM.2013.06.046

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Jie, X., Cao, R., & Li, L. (2014). Emotion recognition based on the sample entropy of EEG. *Bio-Medical Materials and Engineering, 24*(1), 1185–1192. https://doi.org/10.3233/BME-130919

Zhang, Y., Cheng, C., & Zhang, Y. (2021). Multimodal emotion recognition using a hierarchical fusion convolutional neural network. *IEEE Access, 9*, 7943–7951. https://doi.org/10.1109/ACCESS.2021.3049516

Zheng, W. L., Liu, W., Lu, Y., Lu, B. L., & Cichocki, A. (2019). EmotionMeter: A multimodal framework for recognizing human emotions. *IEEE Transactions on Cybernetics, 49*(3), 1110–1122. https://doi.org/10.1109/TCYB.2018.2797176