EXPLAINING PHYSIOLOGICAL AFFECT RECOGNITION WITH OPTIMIZED ENSEMBLES OF CLUSTERED EXPLAINABLE MODELS

Wei Shiung Liew[1], Chu Kiong Loo[2*]

[1,2]Faculty of Computer Science and Information Technology, Universiti Malaya,
50603 Kuala Lumpur, Malaysia

Email: liew.wei.shiung@siswa.um.edu.my[1], ckloo.um@um.edu.my[2*] (corresponding author)

DOI: https://doi.org/10.22452/mjcs.vol35no4.4

ABSTRACT

Affect recognition tasks involving physiological signals are difficult to generalize across a large population due to low signal-to-noise ratio and limited data availability. In addition, the use of deep learning models makes it difficult to determine the cause-and-effect between physiological affect and labeled affect. This work addresses the following issues: uneven distribution and noisy data were addressed using K-Means-SMOTE and Fuzzy ART (FA). The clustered hyper-rectangles were extracted from the FA topology and fitted to an Explainable Boosting Machines ensemble using the Easy Ensemble strategy. The hyper-parameters of the overall methodology were tuned using genetic algorithms for improved generalization. The proposed method was tested using three publicly available affect recognition datasets: DEAP, DREAMER, and AMIGOS. Step-by-step benchmarks showed that combining techniques achieved good generalization and generated explainable information correlating physiological features to affective labels.

Keywords: Affect Recognition, Classifier Ensemble, Explainable, Genetic Algorithms, Physiological Signals

1.0 INTRODUCTION

The ubiquity of wearable electronics has led to diverse applications for monitoring user biometrics in the long term and converting the data into usable information. Standard wearable devices such as smartwatches have integrated sensors for tracking users' cardiac activity and sleeping patterns. In contrast, more advanced devices can track body temperature, blood pressure, and oxygen saturation [1]. Applications then transform the measured biometrics into usable information such as overall fitness, risk of chronic diseases, or early-onset detection using proprietary algorithms with varying levels of accuracy [2].

One application in the field is human affect monitoring. Studies often use the term "affect" interchangeably with emotions and moods [3] or as an overarching category that encompasses adjacent concepts such as stress, or social cues [4]. Affect recognition uses a sensor to measure biometrics such as physiological signals or facial expressions. Then, an algorithm translates the measurements into affective information such as discrete emotion labels. Real-time affective information can be used for sophisticated human-robot interactions [5] [6] or gauging user preferences for targeted advertisements [7] [8].

Obtaining well-defined explanations from datasets with high ambiguity is a difficult proposition [9]. Affect recognition datasets typically model ground truth using affective descriptors that cannot capture the full scope of the human affect [10]. In addition, it is difficult to create a universal predictive model with inter-individual differences in physiological affect responses. This problem may be mitigated with a large enough sample size but is often limited by time and resource constraints. Three popular affect recognition datasets for example, DEAP [11], DREAMER [12], and AMIGOS [13] have less than 50 participants.

Recent advances in deep learning make use of convolutional neural networks (CNNs) for affect recognition [14] [15]. Instead of manually computing features, the physiological signals were converted into spectrograms and directly used as inputs into CNNs. This method bypasses the need for prior knowledge required for feature computation and minimizes information loss from the process. However, the use of CNNs adds a layer of abstraction and makes it difficult to discern the causal relationship between the input features and the affective output labels. Transparent models such as decision trees can produce a set of causal relationships but are vulnerable to overfitting and generalize poorly in response to new inputs.

Dataset pre-processing covers various techniques to prepare the data samples before training a model. Feature extraction and clustering, for example, reduce the dimensionality of the dataset for faster computation. Clustering techniques reshape the topology of the data distribution by grouping highly-similar samples together. Data redundancy is reduced by taking the mean of clusters to represent multiple similar samples. Self-organizing neural network models such as Adaptive Resonance Networks (ART) have built-in clustering abilities that naturally form topologies. Whether the topology is a good representation of the original dataset depends on the network's hyper-parameters. Therefore, parameter tuning is necessary to balance having a representative topology and minimizing redundancies.

This work proposes a framework for generating explainable information from physiological signals for affect recognition applications. The framework encompasses three techniques. Firstly, dataset pre-processing to balance data distribution using data resampling and self-organized neural networks for clustering. Secondly, fitting the processed dataset to an explainable classifier model with emphasis on good generalization performance. Thirdly, generating explainable information from the fitted model that explains the causal relationship between the physiological data and the affect labels.

This work is structured as follows. Chapter 2 reviews the state of art for explainable techniques used for affect recognition. Chapter 3 details the overall framework and the individual techniques used in the framework; Fuzzy ART, K-Means-SMOTE, Explainable Boosting Machine, Easy Ensemble, and Genetic Algorithms. Chapter 4 describes the affect recognition datasets used for benchmarking the performance of the proposed methodology and the experiment methodology. Chapter 5 goes into detail the performance metrics obtained from the experiment as well as the explainable information generated with regards to affect recognition. Finally, Chapter 6 concludes our findings.

2.0 REVIEW OF EXPLAINABLE AFFECT RECOGNITION

A common bottleneck for classification tasks involving physiological signals is the feature extraction technique. While useful for reducing data dimensionality and speed up computation, feature extraction may cause loss of useful information. In recent years, convolutional neural networks (CNNs) were used as an alternative for feature extraction by taking in entire physiological signals without the need for pre-processing. Lin et al. [16] fitted separate physiological signals to individual CNNs and observed the outputs from each network to determine which physiological signal contributes the most towards affect recognition. CNNs are black-boxes however, and this method does not provide the causal relationship between the physiological signals and the studied affect, making it difficult to explain why a particular physiological signal is important. Models such as decision trees provide more transparency but often produce less accurate results.

In addition to the abstract nature of the physiological signals, classifying the information according to abstract concepts such as affect is difficult when the nature of affect is not very well defined. Simplifying affect into a simpler dimensional system such as the Arousal-Valence metrics may produce lower emotional granularity [17], making it difficult to distinguish different emotional states.

Affect recognition datasets with a small sample size may have significant inter-individual differences that make it difficult for classifiers to generalize across a large population [19]. Self-organizing mapping (SOM) is a technique that can be applied to the distribution of the data samples and reduce inter-subject variance [20] in several affect recognition studies involving physiological signals [21] [22] [23]. Fuzzy ART (FA) [24] is useful for classifying noisy physiological data [25] [26] [48] for its self-organizing ability [27]. However, ART-based networks are statistically inconsistent due to dependency on training order and hyper-parameter settings. Therefore, optimizing the training order and hyper-parameter settings is necessary to achieve the best performance.

A technique to address uneven data distribution in affect recognition datasets is by generating synthetic data samples, i.e., using SMOTE [31]. This technique has been used in applications involving noisy [32] and sparse [33] samples, as well as multimodal physiological signals for affect recognition [34]. In some cases however, SMOTE-augmented datasets produced lower generalization when the data distribution is significantly imbalanced [34] [35] [36]. The K-Means-SMOTE technique was developed to address the imbalance issues, although it is still parameter-dependent in order to produce good synthetic samples.

Hyper-parameter tuning is necessary to obtain better results than default parameter settings. It is often difficult to predict the effect of a hyper-parameter on a model, especially for applications with complex fitness functions. Optimization is a process to search for the best result by trial-and-error testing repeatedly for different hyper-

parameter settings. Genetic algorithms (GAs) utilize a biologically-inspired approach for optimization and have been used in several studies for optimizing the performance of FA [28] [29] [30] [47].

Regardless, there is a limit on how much a single model can be optimized without overfitting. Classifier ensembles were often used as a work-around [45] [46]. Ensembles are able to achieve performance that exceeds that of single classifiers by leveraging the strengths of its component classifiers cooperating with each other. A model that is weak when classifying one aspect may be compensated by another model that is strong in that area. While individual classifiers no longer needed to be optimized as much, finding the best combination of weak classifiers for an ensemble is yet another optimization problem that can be solved using GA [28] [30]. However, this additional optimization step will further increase the computation time. In addition, using GA for optimizing the performance of individual classifiers often produce homogeneous models that are ineffective when combined into an ensemble. The Easy Ensemble [43] however is an ensemble technique that generates heterogeneous models, each trained on a different under-sampled section of the dataset, and is able to achieve good ensemble performance even with weak component models.

To summarize, affective recognition from physiological signals is difficult to achieve due to the abstract nature of the studied affect, hampering efforts to generate consistent and reliable explainable information by using transparent models such as decision trees. In addition, the data samples collected from physiological signals contain noise from multiple sources including feature extraction, inter-individual differences, and the methodology for evoking affective states and measuring physiological signals. All of which may produce a dataset with uneven distribution that will negatively affect performance of classifiers. The uneven data distribution may be mitigated using clustering with FA and oversampling using K-Means-SMOTE, but requires significant hyper-parameter tuning to obtain a good cluster topology. GA can be used for hyper-parameter tuning but requires significant computation time to achieve good results. Combining multiple models into a classifier ensemble may reduce the need for optimized models but requires heterogeneous models which is difficult to obtain when using GA for optimization. The Easy Ensemble technique however can create ensembles of heterogeneous models by training identical models on different under-sampled datasets.

3.0 EXPLANATION GENERATION WITH OPTIMIZED ENSEMBLES OF EXPLAINERS

A methodology is proposed for generating explainable information correlating physiological measurements to human affective states, as shown in Figure 1. The methodology can be divided into three broad stages. Firstly, dataset distribution was balanced using a combination of K-Means-SMOTE to generate synthetic samples, Fuzzy ART for clustering, and Easy Ensembles for a combination of undersampling and ensembling. Secondly, the classification performance is optimized using genetic algorithms. Thirdly, explainable information is generated using an explainer model (Explainable Boosting Machine) and SHAP scores.

3.1 Clustering with Fuzzy ART

Physiological datasets often contain noise and artifacts that negatively affect accurate classification. Sources of noise include inexact placement of electrodes, involuntary movements, and high-frequency noise. Clustering is a machine learning technique for grouping many data points into approximated clusters. Individual data points that are similar to each other are grouped closer than dissimilar data points. The cluster center or centroid mean is computed as the aggregate of all data points within the cluster. Two objectives are achieved by taking the centroid means as the representations of the larger dataset. Redundant data points are removed, making it more efficient to iterate through the centroid means. Outlier data points are easily identified as separate from other clusters.

Clustering can be conducted using techniques including K-Means [37] or by using prototype-based classifiers such as ARTMAPs [38]. The former uses algorithmic methods to segment a given population of data points into segments or clusters. The latter develops a topology of prototypes over time in response to a stream of data points during the training process. Each prototype represents a group of sufficiently similar data points and is equivalent to a cluster mean. The topology of prototypes can then be used as a highly compressed representation of the given data points. The topology's shape depends on many factors, including the sequence of data points during training, the similarity metric, and the prototype update method.

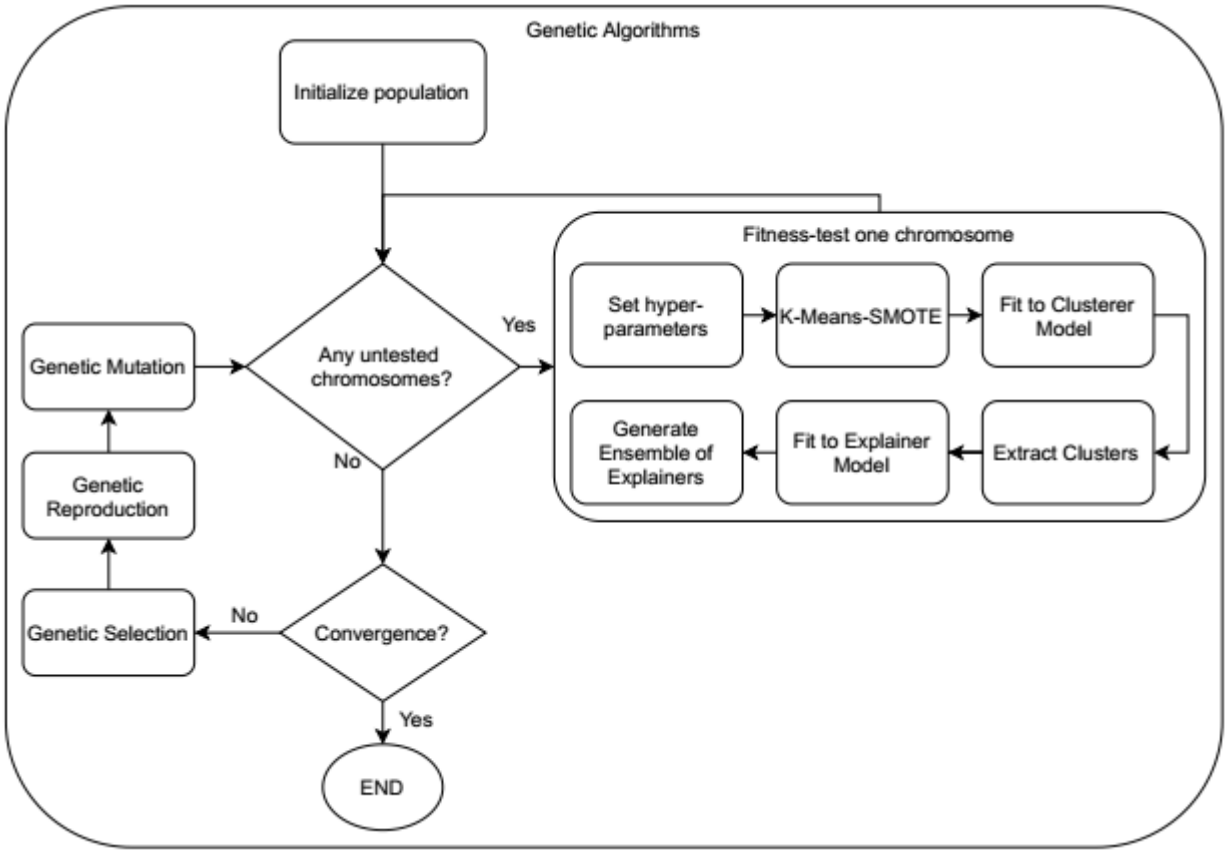

Fig. 1: Overview of proposed methodology.

The Fuzzy ART (FA) [24] is a supervised learning network combining adaptive resonance theory mapping networks (ARTMAP) with fuzzy logic for encoding fuzzy prototypes of data points presented during training. The operations of the FA are summarized as follows:

1. Given an M-dimensional input vector $x = \{x_1, \dots, x_M\}$
2. The input is complement-coded: $\tilde{x} = [x, 1 - x]$. Assuming the input vector x had already been normalized to [0,1] range.
3. Compute the fuzzy membership values between $\tilde{x}$ and existing prototypes w_j for $j \in [1, J]$ where J is the number of existing prototypes in the FA topology:

$$T_j = \frac{|\tilde{x} \wedge w_j|}{\alpha + w_j} \tag{1}$$

   where $\alpha \in [0,1]$ is the fuzzy choice parameter.
4. Hypothesis testing is computed in descending order of fuzzy membership scores T_j:

$$\frac{|\tilde{x} \wedge w_j|}{\tilde{x}} > \rho \tag{2}$$

   where $\rho \in [0,1]$ is the baseline vigilance parameter.
5. When an existing prototype w_j fulfils the vigilance equation in Equation (2), the prototype is updated:

$$w_j^{new} = (1 - \beta) * w_j^{old} + (\beta) * (\tilde{x} \wedge w_j^{old}) \tag{3}$$

   where $\beta \in [0,1]$ is the learning rate parameter.
6. Otherwise if no prototype w_j fulfils the vigilance condition, a new prototype is created:

$$w_{J+1} = \tilde{x} \tag{4}$$

The parameters governing the behavior are the baseline vigilance, fuzzy choice, and learning rate. **Baseline vigilance** $\rho \in [0,1]$ sets the threshold matching score between a prototype and the input vector to determine whether to assign the vector to a new prototype or to activate and update an existing prototype. **Fuzzy choice**

$\alpha \in [0,1]$ is used for selecting prototype candidates based on their similarity to the input. **Learning rate** $\beta \in [0,1]$ is the magnitude of change propagated by the input to the activated prototype.

The combination of Equations 3 and 4 means that the FA is somewhat vulnerable to the sequence of training samples. Presenting the training samples in a different order will produce different results [29] [39]. Tuning the hyperparameters and the training order may significantly improve generalization. However, this is a computationally expensive process, and optimal hyper-parameter settings for one dataset may not be optimal for other datasets.

3.2 Resampling with K-Means-SMOTE

Another common problem among non-synthetic datasets is the imbalanced distribution of class labels. The classifier will generalize poorly for other classes and biases prediction metrics when one class label is over-represented.

Resampling techniques adjust the distribution of data samples. For instance, under-sampling removes over-represented classes and oversampling clones under-represented classes. However, when applied to highly-imbalanced datasets, both resampling techniques have downsides. Under-sampling may discard potentially informative samples from the majority classes. Oversampling a severely imbalanced dataset may overfit the model and negatively impact its ability to generalize to novel minority samples.

Synthetic Minority Oversampling Technique (SMOTE) [31] is a variation of the oversampling technique that generates synthetic samples that are not exact copies of existing minority samples, thus avoiding overfitting. Using SMOTE however amplifies noise and ambiguates the decision function between majority and minority samples. K-Means-SMOTE [40] addresses the downsides of SMOTE using a combination of k-means clustering, filtering, and oversampling.

Given a dataset of samples, k-means clustering assigns each sample to one of k groups based on their proximity to each other. The clustering algorithm iterates two basic instructions: firstly, assign a data sample to the nearest k cluster centroid. Secondly, update the k cluster centroids to be the mean of all data samples assigned to them. The k parameter determines the number of clustered centroids and how accurately they map to the original dataset samples.
After clustering, the filtering step chooses which clustered centroid will be oversampled using SMOTE and how many generated samples to create from each centroid. Since the overall goal is to reduce class imbalance, the filtering step selects clusters that are minority-dominant, using an imbalance class ratio computed as:

$$\text{Imbalance Ratio (ir)} = \frac{\text{number of samples from minority class} + 1}{\text{number of samples from majority class(es)} + 1} \tag{5}$$

Clusters with imbalance ratios above a certain threshold, i.e. $ir_t \geq 1$, are selected as candidates for oversampling. This parameter was set to 1 by default. Next, the filtered clusters are assigned sampling weights to denote how many synthetic samples to generate using SMOTE. Sampling weights are algorithmically computed depending on the density of the cluster compared to the density of all other clusters:

1. For each filtered cluster f, compute the Euclidean distance matrix between all minority samples in f.
2. Compute the mean within-cluster distance from the distance matrix.
3. Compute the cluster density as:

$$\text{density}_f = \frac{\text{minorityCount(f)}}{\text{averageMinorityDistance(f)}^m} \tag{6}$$

   Where m is the number of features.
4. The sparsity of the cluster is computed by taking the inverse of the cluster density.
5. The sampling weight of cluster f is computed as the cluster's sparsity score divided by the sum of all cluster's sparsity scores.

The sampling weight of each cluster can then be multiplied by the number of samples to be generated to obtain the number of samples to be generated from that particular cluster.

Oversampling using SMOTE is then performed for each cluster. Each synthetic sample is generated by interpolating the new sample as a random point in a straight line between l randomly-selected minority samples in a cluster. The SMOTE hyper-parameter l determines how many minority samples are selected for interpolating the new synthetic sample.

There are several hyper-parameters of note: the number of clusters k for k-means clustering, the imbalance ratio threshold ir_t, the total number of synthetic samples n to generate using SMOTE, and the number of nearest neighbours l for SMOTE to generate a new synthetic sample. ir_t was set to the default value of 1 and n were determined during the experiment so that the number of minority class samples were equal to the number of majority class samples.

The optimal number of clusters for k-means was determined by testing for several values of $k \in [2,20]$. After performing k-means clustering for one value of k, the average Silhouette coefficient was computed. A Silhouette score S was computed as:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{7}$$

Where $S(i)$ is the Silhouette for a data point i, $a(i)$ is the average Euclidean distance between i and all other data points within the same cluster, and $b(i)$ is the average Euclidean distance between i and all other data points outside the cluster. The average Silhouette score was computed by averaging the Silhouette of all data points. The optimal number of clusters k was selected with the highest average Silhouette score, indicating a good separation between clusters.

The remaining hyper-parameter l was left to be optimized using genetic algorithms.

3.3 Easy Ensemble

Classifiers work best when given datasets where the labeled clusters are easily distinguished. However, given clusters that have significant overlap, data points that fall in the middle of one or more overlapping clusters are ambiguous, and difficult to assign the correct label. For this matter, training a single classifier model with a dataset containing a complex and overlapping topology may lead to less-than-optimum generalization accuracy. Ensembles of classifiers address this problem by training multiple classifier models on random samplings of the dataset. Each classifier thus contains only a small part of the overall topology, i.e., "weak learners." While individually, each weak learner does not generalize well, combining multiple heterogeneous weak learners may produce better outcomes.

The Easy Ensemble [43] (EE) can be described as a method for generating "ensembles of ensembles". Given a dataset consisting of minority class samples N and majority class samples P. Random undersampling first selects a balanced subset consisting of equal numbers of minority and majority samples: $|P_i| = |N_i|$. The subset was then used for training a weak learner H_i and evaluated to compute its weight α_i. The process was repeated to generate T independent weak learners combined in an ensemble. The number of weak learners T is ideally set to as low as possible while maximizing the overall generalization score of the ensemble. The pseudocode is given in Algorithm 1.

Algorithm 1 EasyEnsemble

procedure EE
  $MaxEnsembleSize = T$
  Minority class samples P, majority class samples N, where $|P| < |N|$
  $i = 0$
  if $i < T$ **then**
    $i \leftarrow i + 1$
    Random undersampling so that $|P_i| = |N_i|$
    Fit explainer H_i with samples P_i and N_i
    $\bar{H}_i(x) = sgn(H_i(x) - \theta_i)$ where θ_i is a threshold
  Output of the ensemble:
  $\bar{H}(x) = sgn(\sum_{i=1}^{T} \alpha_i H_i(x) - \sum_{i=1}^{T} \theta_i)$

3.4 Optimizing Hyper-Parameters with Genetic Algorithms

Hyper-parameter tuning incrementally adjusts values to achieve better performance from the models. However, tuning is a time-consuming and resource-intensive process given a large number of hyper-parameters. Optimization techniques provide a more efficient alternative to discover optimal hyper-parameter settings.

Genetic algorithm (GA) is a search algorithm inspired by biological evolutionary processes. Given a problem to be solved, GA initializes a population of potential solutions and scores them based on their performance. GA searches through the solution space using a combination of genetic selection, reproduction, and mutation. Genetic selection forces the algorithm to focus on high-fitness solutions and away from low-fitness solutions. Genetic reproduction generates slightly different variants of known solutions to explore the vicinity of the solution space for more optimal solutions. However, the combination of selection and reproduction alone may trap the algorithm to examine only the surroundings of known local optima. Genetic mutation introduces randomization and forces the algorithm to explore unknown areas.

In the proposed framework, the "explainer" or "weak learners" is the base classifier model and can be replaced by almost any classifier method. In this experiment, an Explainable Boosting Machine (EBM) [41] [42] was used as the explainer. The EBM is a transparent model designed to provide informative and intelligible data while having state-of-the-art generalization.

Each chromosome consists of one possible combination of hyper-parameter values:

- **Sample importance**. Assuming a dataset consisting of N data samples, sample importance was represented by a vector $\{s_1, \ldots, s_N\}$ where $s_n \in [0, 1]$. Samples were presented for training from the highest sample importance to the lowest.
- **Sample importance threshold**. Less important data samples could be excluded from training without significantly impacting the model's performance. This hyper-parameter is set to a value in the range $S_t = [0, 1]$. During a cross-validation training stage, samples with importance scores below S_t were excluded from training. This parameter does not affect samples in the testing sets.
- K-Means-SMOTE is a technique for generating new synthetic samples using K-Means clustering. The K value sets the **number of nearest neighbors** for averaging a synthetic sample and was assigned to a range [3, 10].
- **Clustering hyper-parameters**. The behavior of the FA was governed by the choice parameter, learning rate, and baseline vigilance. Each variable was set to a range (0, 1].
- **Explainer hyper-parameters**. The performance of the EBM depended on the number of outer and inner bags, learning rate, and the minimum and maximum sample leaves. All the hyper-parameters were set to an arbitrary range [1, 20] except for the learning rate, set to (0, 1].
- **Ensemble hyper-parameters**. The EE algorithm required the user to set the number of estimators in the ensemble to a range [1, 20].

Fitness testing involved using the listed hyper-parameters in one chromosome to initialize, train, and evaluate the performance of the model. The overall performance of the model, i.e. the accuracy in classifying samples, were used as the fitness score for the chromosome. Each chromosome was fitness tested using a K-Fold cross-validation strategy. The specific cross-validation methods used in the experiment varies by dataset and are listed in Table 1. A typical round of fitness testing for one chromosome is described as follows:

1. Divide data samples into training and testing sets. For each fold, data samples assigned for training were not used for testing and vice-versa.
2. Reorder the data samples in the training sets according to their sample importance score in descending order.
3. Exclude training data with low sample importance scores based on the sample importance threshold parameter.
4. Set the K-value for K-Means-SMOTE and resample the selected training data.
5. Initialize an FA model using the listed clustered hyper-parameters.
6. Initialize an EBM model using the listed explainer hyper-parameters.
7. Initialize an EasyEnsemble model as a Pipeline using the listed ensemble hyper-parameters, with the FA in Step (5) and the EBM in Step (6) as the base estimators.
8. The pipelined EE model is fitted with the resampled data from Step (4). The data is used to train the FA, and the prototypes extracted from the FA are used to train the EBM.
9. Test the EE model using the testing set from Step (1).
10. Repeat Steps (1)-(9) as per the cross-validation strategy. The fitness of the chromosome is computed as the average of all testing scores in Step (9).

The following variables influence the GA's effectiveness to converge towards global optima.

- **Population size** (*popsize*) sets the maximum number of chromosomes to maintain. A large population may have a good distribution throughout the solution space and thus arrive at a global optimum earlier.

However, large populations are also computationally expensive to fitness test. Small populations can be evaluated faster but maybe initialized far away from any global optima.

- **Genetic selection rate** (*selrate*) determines how many high-scoring chromosomes will be carried over to the next generation. If the selection rate is low, fewer chromosomes are carried over while more chromosomes are tested per generation. However, if the selection rate is high, fewer new chromosomes are created per generation, negatively impacting the GA's speed in traversing the solution space.
- **Genetic mutation rate** (*mutrate*) in this experiment serves two functions. When a low-scoring chromosome is discarded, *mutrate* is the probability that a completely random chromosome is generated as a replacement instead of using genetic reproduction to mix and match a variant of existing chromosomes. In addition, *mutrate* is also the probability of a gene within the chromosome being randomized to another value. A high *mutrate* forces the GA to explore further away from already-explored regions.

Convergence signifies when the GA cannot locate any other solutions that outperform the current batch of chromosomes. Convergence was determined by comparing the population's fitness at the current generation compared to the previous generations. Whenever a new global optimum was found ($max(F) > f_{best}$), *selrate* and *mutrate* was reset to fast-track the search process in the neighbourhood of the newly-discovered optimum. Finally, the algorithm was considered to have fully converged when the population fitness stopped improving ($mean(F) < f_{mean}$) and ($max(F) < f_{best}$) after several consecutive generations. Otherwise if time is a constraint, the hyper-parameter ($\text{generation}_{\max} = 50$) sets a limit to the number of optimization generations before terminating the GA. The algorithm and the hyper-parameters used in the experiment are summarized in Algorithm 2.

The proposed GA considers the high computation requirements of the fitness functions by setting the initial population size to a small value. To compensate for the sparse distribution of chromosomes across the solution space, the genetic mutation rate was assigned to a high value at the early stages of the GA for aggressive searching. As the GA converges, population size and genetic selection rates were incremented to retain more high-fitness chromosomes. In contrast, genetic mutation rates were reduced to narrow the search around already-discovered chromosomes.

Algorithm 2 Genetic Algorithm

```
procedure GA
    popsize = 10, selrate = 0.5, mutrate = 0.5
    f_best = 0, f_mean = 0
    converged=False, generation=0, generation_max = 50
    Initialize population P = {C_1, ..., C_popsize}
    Initialize fitness F = {f_1, ..., f_popsize}, where f_n = 0
    if converged=False then
        if C_n is untested for n ∈ [1,popsize] then
            f_n ← fitness test C_n
        if f_best < max(F) then
            selrate ← 0.5, mutrate ← 0.5, f_best = max(F)
        if f_mean < mean(F) then f_mean = mean(F)
        if (max(F) <= f_best) and (mean(F) <= f_mean) then
            Slowly converge: popsize++, selrate++, mutrate--
            if selrate > 1.0 then selrate ← 1.0
            if mutrate < 0.1 then mutrate ← 0.1
        if (selrate >= 1.0) or (generation >= generation_max) then
            converged=True
        else Do Genetic Selection, Reproduction, and Mutation
        generation++
        continue
```

4.0 EXPERIMENT

4.1 Affective Datasets

Three affect recognition datasets were used in the experiment: DEAP [11], DREAMER [12], and AMIGOS [13]. Each dataset was created by recording physiological signals of participants undergoing affective stimulus by

watching selected video clips. "Ground truth" affect information was then obtained by polling the participants on their subjective affective states using standardized metrics such as the Arousal-Valence scales [44]. The data collection and processing methods of the three datasets were broadly similar, with a few key differences summarized in Table 1.

Participants were stimulated by viewing audio-visual clips with significant emotional content to evoke distinct affective states. Participants of a previous study viewed a large set of video clips who then scored their subjective emotions using the Arousal-Valence scales. Then for the data collection experiment, a smaller subset of video clips was selected from the extreme ends of the Arousal-Valence scale. The four types of video clips were Low-Arousal, High-Arousal, Low-Valence, and High-Valence to maximize affect stimulus for Arousal and Valence, respectively. Physiological signals were recorded from participants viewing the selected video clips. After each video clip, a cooling-off period was enforced to return their physiological affect to baseline and for the participants to score their subjective affect.

The primary physiological signals of interest in all three datasets were electroencephalogram (EEG) and cardiac-related signals such as electrocardiogram (ECG) and blood volume pulse (BVP). Galvanic skin response (GSR) was measured in the DEAP and AMIGOS studies. Finally, DEAP also collected other peripheral signals such as electromyogram (EMG), electrooculogram (EOG), respiration, and skin temperature.

Access to the datasets was provided from the respective research groups on request. The datasets consisted of the physiological signal recordings annotated with subjective affect scoring, participant ID, and video ID. The physiological signals were pre-processed before conducting feature extraction following the methodologies outlined in their respective publications to recreate the affect classification experiments from the datasets' respective authors.

Table 1: Comparison of key differences between the three physiological affect recognition datasets

Dataset	**DEAP [11]**	**DREAMER [12]**	**AMIGOS [13]**
Number of participants	32	23	40
Stimuli	40 one-minute music videos	18 one-minute music videos	16 short (< 4 mins) and 4 long (< 14 mins) movie clips
Signals	EEG, BVP, GSR, EMG, EOG, Respiration, Skin temperature	EEG, ECG	EEG, ECG, GSR
Segmentation	Last 30 seconds of stimulus recording	Last 60 seconds of stimulus recording	Stimulus recordings divided into 20-second segments
Signal pre-processing	Baseline features obtained from 5 seconds before the experiment and subtracted from stimuli features	Baseline features obtained from last 4 seconds of control session. Stimuli features are normalized by dividing by baseline features	Stimuli features normalized to [-1,+1] range by feature, recording session, and participant
Affect Scoring	Arousal, Valence, Dominance, Liking on continuous [1,9] scale	Arousal, Valence, Dominance on discrete [1,5] scale	Arousal, Valence, Dominance, Liking, Familiarity on continuous [1,9] scale
Binary label threshold	5.0	3.0	Median score
Binary label imbalance	0.736 (Arousal), 0.807 (Valence)	0.776 (Arousal), 0.649 (Valence)	0.748 (Arousal), 0.937 (Valence)
Cross-validation strategy	Leave-one-participant-out	Ten-fold cross-validation with stimuli grouped into ten folds by videoID	Leave-one-participant-out

Two affect classification tasks were conducted for each dataset: Binary Arousal classification and Binary Valence classification. The subjective affect scores were discretized into Low and High classes using a threshold. For instance, the DEAP and DREAMER datasets divide the affect scores into Low or High labels by setting a threshold to the middle of the scale (5.0 and 3.0, respectively). In contrast, the AMIGOS dataset used median scores as the

threshold. The class imbalance score in the table is the ratio of the number of infrequent class label to the number of frequent class label. A score of 1.0 indicates that the Low and High classes have an equal number of data samples, while a score close to 0 indicates significant class imbalance. Training and testing were then conducted using the cross-validation scheme following the datasets' publications. DREAMER used a ten-fold cross-validation strategy by dividing the samples into ten groups according to their video IDs. DEAP and AMIGOS used leave-one-participant-out cross-validation.

5.0 RESULTS AND DISCUSSION

5.1 Affect Classification

Table 2 compares the performance metrics of the original publications of the three datasets in comparison to the findings of this study. There are six classification tasks: Binary Arousal and Binary Valence classification for the datasets DEAP, DREAMER, and AMIGOS. All methods for each task used the same feature extraction and normalization, cross-validation strategies, and performance metrics as listed in Table 1.

Several acronyms are used in the table to indicate which techniques were used. "EBM" denotes using explainer model "Explainable Boosting Machine" on the whole dataset without prior clustering or hyper-parameter optimization. The prefix "GA" indicates the use of genetic algorithms for hyper-parameter optimization. The prefix "FA" is an indicator for dataset pre-processing. K-Means-SMOTE was used for data resampling and Fuzzy ART for clustering before training and testing the explainer models. The suffix "EE" indicates the use of Easy Ensembles for combining multiple fitted models.

Table 2: Comparison of performance metrics with/without hyper-parameter tuning (GA), data pre-processing (FA), or ensembling (EE). Performance metrics reported using F1-score

Experiment	DEAP		DREAMER		AMIGOS	
	Arousal	Valence	Arousal	Valence	Arousal	Valence
Baseline [11] [12] [13]	0.616	0.647	0.575	0.521	0.564	0.560
EBM	0.491	0.500	0.539	0.472	0.524	0.557
GA-EBM	0.504	0.517	0.558	0.544	0.548	0.569
FA-EBM	0.499	0.491	0.541	0.467	0.546	0.560
GA-FA-EBM	***0.668***	***0.676***	0.599	0.587	0.597	0.611
GA-FA-EBM-EE	0.662	0.673	***0.669***	***0.597***	***0.606***	***0.629***

"Baseline" refers to the methods used by the respective datasets' authors for classification. Koelstra et al. [11] and Correa et al. [13] used a Gaussian Naive Bayes classifier with Fisher's linear discriminant for feature selection. Katsigiannis et al. [12] used a Support Vector Machine with Radial Basis Function. From Table 2, using EBM for the same classification tasks without data pre-processing or hyper-parameter optimization produced lower F1-scores than the baseline methods. However, with hyper-parameter optimization, two out of six tasks achieved higher scores than the baseline: GA-EBM for DREAMER Valence and AMIGOS Valence.

For "FA-EBM," K-Means-SMOTE was used for resampling the datasets before clustering with FA. The prototypes embedded in the fitted FA were then extracted to be fitted to the explainers. Only one out of six tasks produced better or equal generalization than the baseline: FA-EBM for AMIGOS Valence classification. When combined with hyper-parameter optimization, however, all classification tasks outperformed baseline.

Lastly, ensembling the pre-processed and optimized explainer models showed an improvement over non-ensembled models in four of the six classification tasks. In the case of GA-FA-EBM-EE for DEAP Arousal and Valence classification, ensembling produced lower generalization than individual models.

5.2 Explaining Feature Importance

For the GA-FA-EBM-EE method in Table 2, the chromosomes with the highest fitness scores from each of the six classification tasks were selected. Ensembles of EBMs were then created according to the hyper-parameters listed in the chromosome. Instead of following the cross-validation strategies used during fitness testing, the ensembles were created in one pass as follows:

1. Data samples were reordered according to their sample importance scores in descending order.
2. Data samples with importance scores below the threshold parameter were excluded.

3. The K-value was set for K-Means-SMOTE. The selected data samples were then resampled.
4. An FA was initialized using the clusterer hyper-parameters.
5. An EBM was initialized using the explainer hyper-parameters.
6. An Easy Ensemble model was initialized as a Pipeline using the ensemble hyper-parameters and using the FA and EBM from Steps (4)-(5) as the base estimator.
7. The pipelined EE model was then fitted with the resampled data from Step (3). The FA model was fitted with the data, and the prototypes were extracted for training the EBM.

Each ensemble was then used as a kernel for computing SHAP scores [18] for all data samples from the respective datasets. Computing sample and feature importance are model-dependent, leading to different results when a different fitted model was used to compute SHAP scores. Obtaining a high SHAP score is not necessarily an indicator of objective importance but have to be considered in context of the generalization performance of the classifier model from which the SHAP score was derived from. Therefore, obtaining a high SHAP score in relation to a model with good generalization performance is preferable than getting a high SHAP from a low-performing model.

5.2.1 Effects of Resampling and Clustering

The effects of resampling and clustering on the data samples were observed by taking a fitted model as a kernel for computing SHAP scores for all the samples in the dataset. Two fitted models are compared: "Unprocessed" is an EBM which was fitted with the unmodified data samples from a dataset, and "Processed" is a GA-FA-EBM which was fitted with data samples that have been sorted based on the sample importance score and filtered by an importance threshold, followed by resampling with K-Means-SMOTE and clustering with FA. The EBM hyper-parameters for both models were set to the default instead of using their respective optimized values.

Table 3: Importance of physiological signals represented using normalized SHAP scores. "Unprocessed" refers to SHAP scores computed from original data samples. "Processed" refers to SHAP scores computed from resampled and clustered hyper-rectangles extracted from optimized FA topology

Experiment	DEAP		DREAMER		AMIGOS	
	Arousal	Valence	Arousal	Valence	Arousal	Valence
EEG (unprocessed)	0.593	0.562	0.320	0.345	0.433	0.401
EEG (processed)	0.591	0.583	0.337	0.330	0.439	0.400
Δ	-0.002	+0.021	+0.017	-0.015	+0.006	-0.001
ECG (unprocessed)	0.172	0.120	0.679	0.654	0.458	0.473
ECG (processed)	0.168	0.101	0.662	0.669	0.456	0.475
Δ	-0.004	-0.019	-0.017	+0.015	-0.002	+0.002
GSR (unprocessed)	0.048	0.060	-	-	0.108	0.125
GSR (processed)	0.052	0.085	-	-	0.103	0.124
Δ	+0.004	+0.025	-	-	-0.005	-0.001
EMG/EOG (unprocessed)	0.115	0.181	-	-	-	-
EMG/EOG (processed)	0.110	0.155	-	-	-	-
Δ	-0.005	-0.026	-	-	-	-
Resp (unprocessed)	0.062	0.067	-	-	-	-
Resp (processed)	0.065	0.065	-	-	-	-
Δ	+0.002	-0.002	-	-	-	-
Temp (unprocessed)	0.006	0.008	-	-	-	-
Temp (processed)	0.010	0.009	-	-	-	-
Δ	+0.004	+0.001	-	-	-	-

Table 3 compares the SHAP scores computed from the two fitted models for the three datasets. SHAP scores were computed individually for each feature and then grouped according to physiological signal. The SHAP scores were normalized so that the sum of SHAP scores for each dataset across all physiological signals sum up to 1.

Number of clusters: Compared to the number of samples in the unmodified datasets, the number of hyper-rectangles extracted from the optimized FA was approximately 6%-14% less across all classification tasks. This may indicate that most of the data samples are considered non-redundant and cannot be easily excluded. As shown in Table 2 when comparing "EBM" to "FA-EBM," the resampling and clustering procedure caused some loss of

important information resulting in lower generalization performance. However, hyper-parameter tuning minimized information loss and produced a significantly better generalization when comparing "GA-EBM" to "GA-FA-EBM."

Feature importance: SHAP scores were computed for all data samples for each of the six classification tasks, using the EBM model as a SHAP kernel for "Unprocessed" samples and GA-FA-EBM model as a SHAP kernel for "Processed" samples. When comparing the SHAP scores of features between the two methods, it was observed that while the individual SHAP scores of features may vary by up to 300%, the normalized proportion of SHAP scores between physiological signal channels (i.e., EEG, ECG) varied by as much as 26% as shown in Table 3.

5.2.2 Explaining Classification for DEAP

Fig. 2 (left) shows the features with the most significant contributions towards the Arousal classification of the DEAP dataset, sorted from most to least important. The left of the axis represents the Low Arousal class, while the right of the axis represents the High Arousal class. The frequency of the dots corresponds to the number of data samples while the distance of the dots from the central axis is relative to the importance of the features, and the color of the dots are the feature values. For instance, the beta band power at the EEG electrodes CP6, FC2, and F7 are important predictors of Low Arousal. The theta band power at the electrode position T7, on the other hand, is more likely to predict High Arousal. The gamma-band power at the F7 electrode, for example, showed that medium-to-high values of the feature are predictors for Low Arousal.

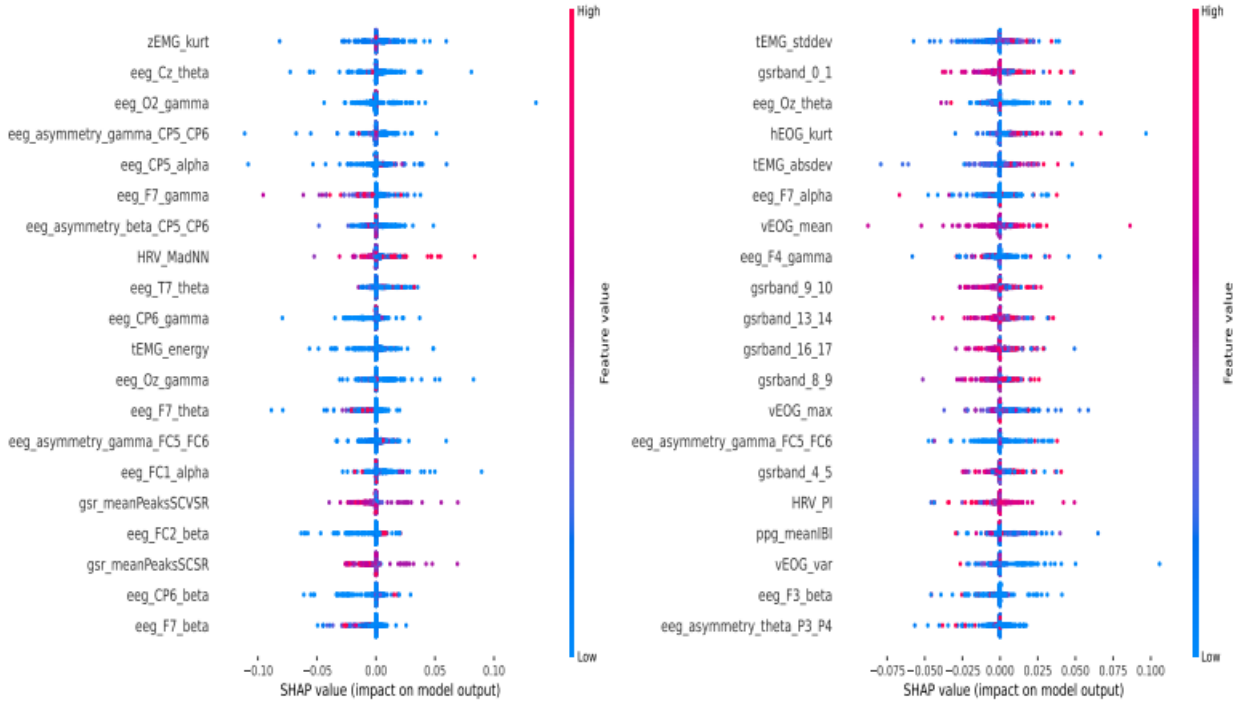

Fig. 2: Significant feature contributions for DEAP Arousal classification (left) and Valence classification (right)

In contrast, low values are predictors of High Arousal. Significant EMG features are the kurtosis of the zygomaticus muscle amplitude and the signal energy of the trapezius muscle. For GSR, the skin conductance slow response and very slow response were significant. Lastly, the mean absolute deviation of the heart rate NN peaks appeared to be an important predictor for High Arousal.

Fig. 2 (right) shows the significant features for the Valence classification of the DEAP dataset. The left of the axis represents the Negative Valence class, while the right of the axis represents the Positive Valence class. Compared to the Arousal classification task, fewer EEG features were significant contributors to the model's predictions. Several GSR frequency band power features were prominent, in addition to EMG and EOG features.

While most of the features were equally important for both Valence classes, a few were exclusive towards Negative or Positive classes. For example, the standard deviation of the trapezius EMG signal (tEMG), the beta band power at the F3 EEG electrode, and the theta band power asymmetry between the electrodes P3 and P4 showed more bias

towards the Negative Valence class. Meanwhile, the kurtosis of the horizontal EOG signal, the maximum amplitude and variance of the vertical EOG signal, and the mean inter-beat intervals of the PPG signal all showed heavy bias towards Positive Valence.

A comparison of the two graphs showed very little overlap. The gamma-band power asymmetry between FC5 and FC6 is the only common significant feature for Arousal and Valence classification. Arousal classification was heavily dependent on EEG frequency band power. On the other hand, Valence classification relied more on GSR frequency band power and EMG/EOG features.

5.2.3 Explaining Classification for DREAMER

Fig. 3 (left) shows the features with significant contributions toward the Arousal classification of the DREAMER dataset. The majority consisted of ECG features, particularly the statistical indices related to the PQRST-wave and ECG frequency band power below 0.6Hz. The delineation of feature values divided by the central axis shows that some features can be divided into Low or High Arousal using a threshold value. For example, low values of ecg_p_range and ecg_s_range were almost exclusively predictors for Low Arousal, while low values for HRV_HF and HRV_HFn were biased towards High Arousal.

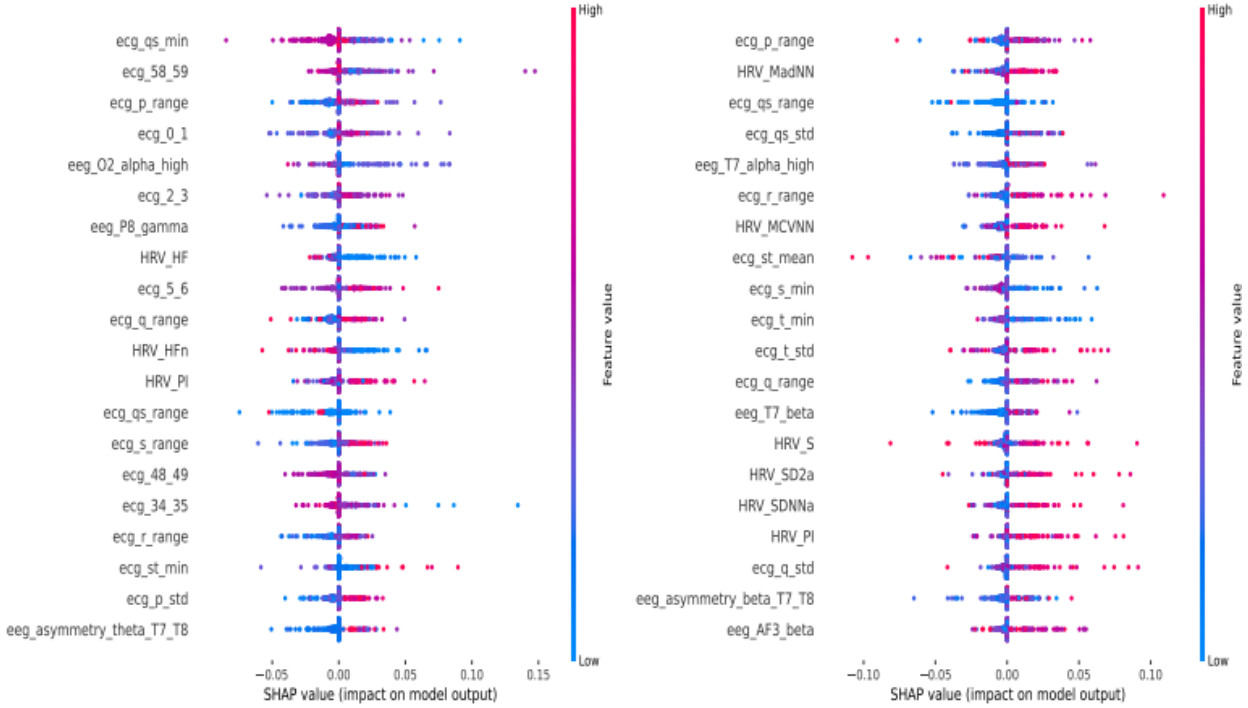

Fig. 3: Significant feature contributions for DREAMER Arousal classification (left) and Valence classification (right)

Fig. 3 (right) shows the features with significant contributions toward the Valence classification of the DREAMER dataset. From the data distribution on the right side of the central axis, most of the significant features biased towards predicting the Positive Valence class consisted of high feature values. ECG and HRV features make up the majority of the significant contributors. Features that were heavily biased towards predicting the Negative Valence class include the interval range of the ECG Q-S waveform, the mean interval of the ECG S-T waveform, and the beta band power of the EEG T7 electrode.

There are several significant features common to both Arousal and Valence classification for the DREAMER dataset: the range of the amplitudes for the ECG P, Q, and R signals, the interval range for the ECG Q-S waveform, and the HRV Porta's Index. In addition, the frequency band power asymmetry between the EEG electrodes T7 and T8 was significant in both classification tasks, albeit theta band power for Arousal classification and beta band power for Valence.

5.2.4 Explaining Classification for AMIGOS

Fig. 4 (left) shows the features with significant contributions toward the Arousal classification of the AMIGOS dataset. The data distribution around the central axis shows that a majority of features were almost exclusively significant predictors of the Low Arousal class: the ECG band power for the frequency ranges 1.0Hz-1.1Hz, 4.4Hz-4.5Hz and 5.5Hz-5.9Hz, the GSR band power for the frequency ranges 0.3Hz-0.4Hz and 0.6Hz-0.7Hz, the EEG gamma-band power asymmetry between AF3 and AF4, the EEG alpha band power asymmetry between T7 and T8, and the EEG gamma-band power at F4. The High Arousal class has fewer exclusive features: the EEG alpha band power at AF4 and the percentage of NN intervals in alternation segments (HRV_PAS).

Fig. 4 (right) shows the features with significant contributions toward the Valence classification of the AMIGOS dataset. Significant predictors of Positive Valence include several HRV features related to NN intervals (i.e., mean absolute deviation (MadNN), total variance of contributions of decelerations (SDNNd)), the ECG band power for the frequency ranges 2.8Hz-2.9Hz, 3.6Hz-3.7Hz, and 3.9Hz-4.0Hz, and the EEG gamma-band power asymmetry between O1 and O2. Significant predictors of Negative Valence include high feature values of HRV_MedianNN and EEG gamma-band power asymmetry between F7 and F8, low feature values of EEG gamma-band power at O2, EEG theta band power at P7.

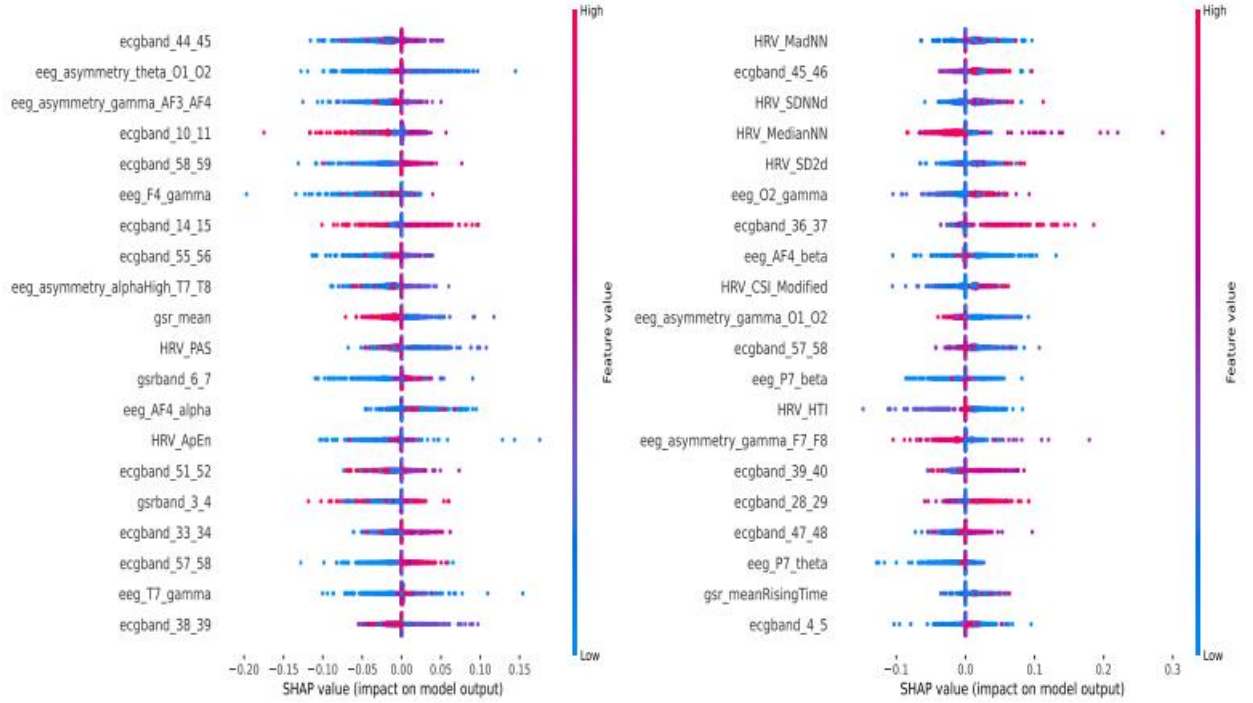

Fig. 4: Significant feature contributions for AMIGOS Arousal classification (left) and Valence classification (right)

The ECG band power for the frequency band 5.7Hz-5.8Hz appeared significant in both classification tasks. Low values for the feature were a predictor for Low Arousal and Positive Valence. Similarly, the EEG band power at AF4 was also significant; the alpha band power for Arousal classification and beta band power for Valence classification. For the EEG band power asymmetry between O1 and O2, the theta band power was a predictor for the Arousal class, while the gamma-band power was a predictor for Positive Valence.

6.0 CONCLUSION

A methodology was proposed to provide an explainable affect recognition model. Dataset pre-processing was conducted using K-means-SMOTE to optimize the distribution of the data samples and Fuzzy ART for clustering and to reduce variability. The clustered data samples were then fitted into an ensemble of explainable classifier models with the help of the Easy Ensemble technique to create a balanced ensemble. The hyper-parameters involved during the entire process were optimized using Genetic Algorithms. The final optimized ensemble of classifiers was then analyzed using SHAP scores to obtain explainable information regarding recognizing affective states from physiological signal features.

The proposed methodology was tested using three publicly available affect recognition datasets. The dataset pre-processing methods were tested by separately training and testing two models, one with the original dataset and one with the processed dataset. The pre-processed data samples were 6-14% smaller than the unprocessed dataset and showed significantly different feature importance scores when analyzed using SHAP. In addition, a series of tests were conducted by selectively removing certain steps in the proposed methodology. It was shown in Table 2 that the combination of all the steps were able to produce the best generalization performance for two of the three datasets. The DEAP dataset on the other hand showed better performance without the need for Easy Ensembles.

REFERENCES

[1] Mamdiwar, S.D.; Shakruwala, Z.; Chadha, U.; Srinivasan, K.; Chang, C.Y.; others. Recent advances on IoT-assisted wearable sensor systems for healthcare monitoring. Biosensors 2021, 11, 372.

[2] King, C.E.; Sarrafzadeh, M. A survey of smartwatches in remote health monitoring. Journal of healthcare informatics research 2018, 2, 1–24.

[3] Ekkekakis, P. The measurement of affect, mood, and emotion: A guide for health-behavioral research; Cambridge University Press, 2013.

[4] McColl, D.; Hong, A.; Hatakeyama, N.; Nejat, G.; Benhabib, B. A survey of autonomous human affect detection methods for social robots engaged in natural HRI. Journal of Intelligent & Robotic Systems 2016, 82, 101–133.

[5] Rani, P.; Sarkar, N.; Smith, C.A.; Kirby, L.D. Anxiety detecting robotic system–towards implicit human-robot collaboration. Robotica 2004, 22, 85–95.

[6] Filntisis, P.P.; Efthymiou, N.; Koutras, P.; Potamianos, G.; Maragos, P. Fusing body posture with facial expressions for joint recognition of affect in child–robot interaction. IEEE Robotics and Automation Letters 2019, 4, 4011–4018.

[7] Okada, G.; Masui, K.; Tsumura, N. Advertisement effectiveness estimation based on crowdsourced multimodal affective responses. Proceedings of the IEEE Conference on computer vision and pattern recognition workshops, 2018, pp. 1263–1271.

[8] Sripian, P.; Anuardi, M.N.A.M.; Yu, J.; Sugaya, M. The implementation and evaluation of individual preference in robot facial expression based on emotion estimation using biological signals. Sensors 2021, 21, 6322.

[9] Weidman, A.C.; Steckler, C.M.; Tracy, J.L. The jingle and jangle of emotion assessment: Imprecise measurement, casual scale usage, and conceptual fuzziness in emotion research. Emotion 2017, 17, 267.

[10] Arce, E.; Simmons, A.N.; Stein, M.B.; Winkielman, P.; Hitchcock, C.; Paulus, M.P. Association between individual differences in self-reported emotional resilience and the affective perception of neutral faces. Journal of affective disorders 2009, 114, 286–293.

[11] Koelstra, S.; Muhl, C.; Soleymani, M.; Lee, J.S.; Yazdani, A.; Ebrahimi, T.; Pun, T.; Nijholt, A.; Patras, I. DEAP: A database for emotion analysis using physiological signals. IEEE Transactions on Affective Computing 2011, 3, 18–31.

[12] Katsigiannis, S.; Ramzan, N. DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. IEEE Journal of Biomedical and Health Informatics 2017, 22, 98–107.

[13] Correa, J.A.M.; Abadi, M.K.; Sebe, N.; Patras, I. AMIGOS: A dataset for affect, personality and mood research on individuals and groups. IEEE Transactions on Affective Computing 2018.

[14] Asghar, M.A.; Khan, M.J.; Amin, Y.; Rizwan, M.; Rahman, M.; Badnava, S.; Mirjavadi, S.S.; others. EEG-based multi-modal emotion recognition using bag of deep features: An optimal feature selection approach. Sensors 2019, 19, 5218.

[15] Cimtay, Y.; Ekmekcioglu, E. Investigating the use of pretrained convolutional neural network on cross-subject and cross-dataset EEG emotion recognition. Sensors 2020, 20, 2034.

[16] Lin, J.; Pan, S.; Lee, C.S.; Oviatt, S. An explainable deep fusion network for affect recognition using physiological signals. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2069–2072.

[17] Arce E., Simmons, A. N., Stein, M. B., Winkielman P., Hitchcock C., Paulus, M. P. Association between individual differences in self-reported emotional resilience and the affective perception of neutral faces. Journal of Affective Disorders 2009, 114, pp. 286-293.

[18] Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4768–4777.

[19] Liu, Y.; Sourina, O.; Nguyen, M.K. Real-time EEG-based human emotion recognition and visualization. 2010 international conference on cyberworlds. IEEE, 2010, pp. 262–269.

[20] Xu, Q.; Nwe, T.L.; Guan, C. Cluster-based analysis for personalized stress evaluation using physiological signals. IEEE journal of biomedical and health informatics 2014, 19, 275–281.

[21] Masulli, P.; Masulli, F.; Rovetta, S.; Lintas, A.; Villa, A.E. Fuzzy clustering for exploratory analysis of EEG event-related potentials. IEEE Transactions on Fuzzy Systems 2019, 28, 28–38.

[22] Ghoniem, R.M.; Algarni, A.D.; Shaalan, K. Multi-modal emotion aware system based on fusion of speech and brain information. Information 2019, 10, 239.

[23] Škrjanc, I. Cluster-volume-based merging approach for incrementally evolving fuzzy Gaussian clustering—eGAUSS+. IEEE Transactions on Fuzzy Systems 2019, 28, 2222–2231.

[24] Carpenter, G.A.; Grossberg, S.; Markuzon, N.; Reynolds, J.H.; Rosen, D.B.; others. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on neural networks 1992, 3, 698–713.

[25] Palaniappan, R.; Raveendran, P.; Nishida, S.; Saiwaki, N. Fuzzy ATRMAP classification of mental tasks using segmented and overlapped EEG signals. 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No. 00CH37119). IEEE, 2000, Vol. 2, pp. 388–391.

[26] Jafarifarmand, A.; Badamchizadeh, M.A.; Khanmohammadi, S.; Nazari, M.A.; Tazehkand, B.M. A new self-regulated neuro-fuzzy framework for classification of EEG signals in motor imagery BCI. IEEE transactions on fuzzy systems 2017, 26, 1485–1497.

[27] Vineyard, C.M.; Verzi, S.J.; Bernard, M.L.; Taylor, S.E.; Dubicka, I.; Caudell, T.P. A multi-modal network architecture for knowledge discovery. Security Informatics 2012, 1, 1–12.

[28] Loo, C.K.; Liew, W.S.; Sayeed, M.S. Genetic ensemble biased ARTMAP method of ECG-based emotion classification. In Intelligent Interactive Multimedia: Systems and Services; Springer, 2012; pp. 299–306.

[29] Yaghini, M.; Shadmani, M.A. GOFAM: a hybrid neural network classifier combining fuzzy ARTMAP and genetic algorithm. Artificial Intelligence Review 2013, 39, 183–193.

[30] Liew, W.S.; Seera, M.; Loo, C.K.; Lim, E. Affect classification using genetic-optimized ensembles of fuzzy ARTMAPs. Applied Soft Computing 2015, 27, 53–63.

[31] Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2002, 16, 321–357.

[32] Kaur, P.; Gosain, A. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In ICT Based Innovations; Springer, 2018; pp. 23–30.

[33] Vanhoeyveld, J.; Martens, D. Imbalanced classification in sparse and large behaviour datasets. Data Mining and Knowledge Discovery 2018, 32, 25–82.

[34] Monkaresi, H.; Bosch, N.; Calvo, R.A.; D'Mello, S.K. Automated detection of engagement using video-based estimation of facial expressions and heart rate. IEEE Transactions on Affective Computing 2016, 8, 15–28

[35] Bang, J.; Hur, T.; Kim, D.; Lee, J.; Han, Y.; Banos, O.; Kim, J.I.; Lee, S.; others. Adaptive data boosting technique for robust personalized speech emotion in emotionally-imbalanced small-sample environments. Sensors 2018, 18, 3744.

[36] Elreedy, D.; Atiya, A.F. A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Information Sciences 2019, 505, 32–64.

[37] Ahmad, A.; Dey, L. A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering 2007, 63, 503–527.

[38] Carpenter, G.A.; Grossberg, S.; Reynolds, J.H. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural networks 1991, 4, 565–588.

[39] Loo, C.K.; Liew, W.S.; Seera, M.; Lim, E. Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms. Neural Computing and Applications 2015, 26, 263–276.

[40] Last, F.; Douzas, G.; Bacao, F. Oversampling for imbalanced learning based on k-means and smote. arXiv preprint arXiv:1711.00837 2017.

[41] Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate intelligible models with pairwise interactions. Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, pp. 623–631.

[42] Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. Interpretml: A unified framework for machine learning interpretability. arXiv preprint arXiv:1909.09223 2019.

[43] Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 2008, 39, 539–550.

[44] Carroll, J.M.; Yik, M.S.; Russell, J.A.; Barrett, L.F. On the psychometric principles of affect. Review of General Psychology 1999, 3, 14–22.

[45] Kobayashi, V. B.; Calag, V. B. Detection of affective states from speech signals using ensembles of classifiers. 2013.

[46] Thiam, P.; Kächele, M.; Schwenker, F.; Palm, G. Ensembles of support vector data description for active learning based annotation of affective corpora. In 2015 IEEE symposium series on computational intelligence, 2015, 1801-1807.

[47] Al-Shboul, B., Faris, H., & Ghatasheh, N. Initializing genetic programming using fuzzy clustering and its application in churn prediction in the telecom industry. Malaysian Journal of Computer Science 2015, 28(3), 213-220.

[48] Palaniappan R, Raveendran P. Cognitive task prediction using parametric spectral analysis of EEG signals. Malaysian Journal of Computer Science. 2001 Jun 1;14(1):58-67.