applied sciences

MDPI

Article

Developing Innovative Feature Extraction Techniques from the Emotion Recognition Field on Motor Imagery Using Brain–Computer Interface EEG Signals

Amr F. Mohamed and Vacius Jusas *

Department of Software Engineering, Kaunas University of Technology, Studentų 50, LT-51390 Kaunas, Lithuania; amr.mohamed@ktu.edu

* Correspondence: vacius.jusas@ktu.lt

Abstract: Research on brain–computer interfaces (BCIs) advances the way scientists understand how the human brain functions. The BCI system, which is based on the use of electroencephalography (EEG) signals to detect motor imagery (MI) tasks, enables opportunities for various applications in stroke rehabilitation, neuroprosthetic devices, and communication tools. BCIs can also be used in emotion recognition (ER) research to depict the sophistication of human emotions by improving mental health monitoring, human–computer interactions, and neuromarketing. To address the low accuracy of MI-BCI, which is a key issue faced by researchers, this study employs a new approach that has been proven to have the potential to enhance motor imagery classification accuracy. The basic idea behind the approach is to apply feature extraction methods from the field of emotion recognition to the field of motor imagery. Six feature sets and four classifiers were explored using four MI classes (left and right hands, both feet, and tongue) from the BCI Competition IV 2a dataset. Statistical, wavelet analysis, Hjorth parameters, higher-order spectra, fractal dimensions (Katz, Higuchi, and Petrosian), and a five-dimensional combination of all five feature sets were implemented. GSVM, CART, LinearSVM, and SVM with polynomial kernel classifiers were considered. Our findings show that 3D fractal dimensions predominantly outperform all other feature sets, specifically during LinearSVM classification, accomplishing nearly 79.1% mean accuracy, superior to the state-of-the-art results obtained from the referenced MI paper, where CSP reached 73.7% and Riemannian methods reached 75.5%. It even performs as well as the latest TWSB method, which also reached approximately 79.1%. These outcomes emphasize that the new hybrid approach in the motor imagery/emotion recognition field improves classification accuracy when applied to motor imagery EEG signals, thus enhancing MI-BCI performance.

Keywords: electroencephalogram; emotion recognition; motor imagery; EEG feature extraction; brain–computer interface

Citation: Mohamed, A.F.; Jusas, V. Developing Innovative Feature Extraction Techniques from the Emotion Recognition Field on Motor Imagery Using Brain–Computer Interface EEG Signals. *Appl. Sci.* **2024**, *14*, 11323. https://doi.org/10.3390/app142311323

Academic Editor: Jing Jin

Received: 25 October 2024
Revised: 18 November 2024
Accepted: 26 November 2024
Published: 4 December 2024

1. Introduction

To explore the complexity of the human brain in terms of electrical signals emitted during cerebral activity, the measurement and monitoring of cerebral activity are performed using several approaches. In this study, we start by providing an overview of brain activity measurement techniques. First is functional magnetic resonance imaging (fMRI), which uses electromagnetic fields to track changes in blood flow, blood volume, and oxygen levels during brain activity imaging; however, its suitability for fast BCI systems is not great, and head movements easily affect the tracking technique [1]. Second is functional near-infrared spectroscopy (fNIRS), which tracks brain metabolism; this technique uses infrared light, but it is limited to the outer layers of the brain due to the light penetration depth [2]. The most important technique involves electroencephalography (EEG) signals; during brain neural activation, electric currents are emitted. EEG is used to detect electrical brain activity, but signals can sometimes be affected by other electrical currents in the body [3].

The brain–computer interface system captures EEG signals through sensors and electrodes integrated into the device. These signals are then digitized and stored in a format suitable for research-ready explorations [4]. The technique commonly used by scientists and neurologists nowadays is EEG. It has essential advantages, which is why it is popular among researchers in this field of study: its non-intrusive nature, the fact that it does not necessitate a surgical procedure, its low cost compared to the other approaches mentioned above, and the fact that it operates entirely through a brain–computer interface placed on the subject's scalp [5].

The importance of the emotion recognition (ER) field is significantly increasing with the number of electronic devices used by individuals; people are spending more time on social media, playing online video games, shopping online, and using other electronic products [6]. The capability of processing human emotions and comprehending them is crucial for studying and investigating emotional recognition procedures with different emotion states (i.e., valence, arousal, happiness, and sadness) and how they influence the subject's interaction with their surroundings and even their decision-making abilities [7].

The motor imagery (MI) field introduces numerous potential applications to improve lives; it helps patients regain the ability to perform tasks they lost due to a debilitating illness, a brain injury, or even a severe accident. Detecting abnormal brain activity in clinical diagnosis can help with epilepsy or seizures, help people restore vision, or even allow a person with physical disabilities to perform motor imagery tasks (such as the imagination of movement, considering the right and left hands, feet, and tongue) and control a complex external device using their brain [8].

There are several challenges facing scientists in the motor imagery field, including some limitations regarding current feature extraction and classification methods in MI tasks. First is the low signal-to-noise ratio (SNR) due to the noisy characteristics of EEG signals (i.e., muscle movements, eye blinks, and electrical interference), making the hunt for valuable features that reflect MI tasks, considering brain activity, hard to discern [9]. Second is the use of traditional features (i.e., power spectral densities and band power), which often face difficulties in capturing the most significant information needed to differentiate between motor imagery tasks, causing lower accuracy [10]. Finally, advanced feature extraction methods and performance classifiers can be computationally demanding, making them difficult to use effectively for motor imagery tasks.

This is where our research can be used in practice, as this study's main idea is the cross-application of both techniques from emotion recognition (ER-EEG) and motor imagery (MI-EEG) tasks, leveraging ER feature extraction techniques to improve MI classification. Our main focus is on EEG signal analysis for motor imagery classification, and any improvements in the classification accuracy have important implications for brain–computer interface (BCI) applications, as BCIs depend on the precise detection of MI tasks to facilitate control over external devices. By improving motor imagery classification accuracy, this study contributes to the BCI field, showing improvements for more reliable and effective BCI systems that can be used by researchers in the field in further explorations.

The current work primarily aims to train and evaluate various machine learning classifiers in addition to feature extraction techniques from the emotion recognition field applied to the motor imagery field using electroencephalography signals. The specific goals are grouped into three parts: first, to adapt emotion recognition feature extraction methods for motor imagery data; second, to assess the performance of these features with various classifiers; finally, to compare the results with traditional motor imagery approaches.

The novelty of the current work is the introduction of ER-based feature extraction to MI tasks. Given the impact and potential to enhance motor imagery classification accuracy, this study benefits brain–computer interface applications for individuals with physically disabilities.

2. Related Work

This section will discuss the classification methods commonly applied in the emotion recognition and motor imagery fields. Additionally, it will cover the feature extraction techniques and preprocessing steps that are typically employed. Then, it will discuss the related work in the motor imagery field, highlighting the key issues tackled in this study.

Commencing with the preprocessing techniques in [7], the ground truth is established by converting the ER-EEG datasets into self-report scales. All the data from the latter went through a four-sequence preprocessing phase: a notch filter, a high-pass filter, downsampling, and the implementation of a common average reference (CAR) montage. Then, the feature extraction techniques were applied. Following the preprocessing phase, the reference article digests five main feature extraction methods: statistical, wavelet, fractal dimensions, Hjorth parameters, and higher-order spectra (HOS), in addition to a combination of all of the latter simultaneously. Finally, regarding performance classifiers, this study employed two primary algorithms: the Gaussian radial basis function support vector machine (GSVM) and the classification and regression tree (CART). As the mentioned study only utilized the binary valence–arousal (positive and negative) plane, the findings indicated that the CART classifier outperformed the GSVM in recognizing emotions from EEG data. Moreover, the fractal dimensions feature set, which includes the Katz [11], Petrosian [12], and Higuchi [13] algorithms, achieved the highest performance classifier accuracy.

After interpreting the motor imagery field, regarding dataset usage, the considered research article [14] underlines one prominent motor imagery-related EEG dataset: the BCI Competition IV 2a set [15]. This dataset involved motor imagery tasks, with four event types representing cues for left-hand, right-hand, foot, and tongue movements. Runs four through nine focused on these motor imagery tasks, while runs one to three were dedicated to recording eye movement. Nine subjects participated in the study, each completing nine runs. The data were sampled at 250 Hz using 25 electrodes, 22 of which recorded EEG signals, while the remaining 3 were used for electrooculography (EOG). Applying feature extraction techniques underscores a pipeline of two main feature extraction methods: common spatial patterns (CSP) with multiscale temporal and Riemannian covariance methods. Regarding classifiers, the support vector machine (SVM) was used with linear, rbf, and poly sub-kernels. The results showed that the multiscale CSP features reached an average classification accuracy of 73.70%, which is better than the current leading method, which achieved 70.60%. During the usage of the Riemannian covariance features, it performed better than CSP, with an accuracy of 74.27%; with more time windows, the accuracy jumped to 75.47%. The latter outcomes were achieved across all nine subjects using all four classes from the BCI Competition IV-2a dataset [14].

All researchers in the current field of study, the MI field, have one primary concern: enhancing MI-BCI performance. Five main improvement procedures can be followed [16]: first is preprocessing, which can be achieved by improving or filtering the signal. The latter can be achieved by reconstructing the signal [17,18], improving the spatial resolution [19], or adding artificial noise [20]. Second is using channel selection by removing redundant and non-task-relevant channels [21] and reducing the device's power consumption [22]. Third is using feature selection, which improves performance by finding the most optimal features. Fourth is dimensionality reduction, which is accomplished by reducing the number of features while retaining as much relevant information as possible. Lastly, combining all the previously mentioned techniques and aiming for general improvement can also achieve higher performance accuracy [16].

The present work being implemented in this study will focus on the three preprocessing filters: notch filter, high-pass filter, and common average reference (CAR) montage without downsampling. The neglect of downsampling was considered, as it would be necessary if we needed to reduce data size for computational efficiency; however, the adopted dataset had a sampling frequency that is sufficient for capturing relevant EEG bands. Feature extraction used statistical features, wavelet analysis, fractal dimensions, Hjorth parameters, higher-order spectra, and a combination of all the previously mentioned

features. Related classifier techniques from the ER-EEG pipeline [7] included GSVM (an SVM classifier with rbf kernel) and CART (regression tree).

An SVM classifier with a polynomial kernel and the LinearSVM classifier were introduced to the MI-EEG pipeline [14] specifically only using CSP features without Riemannian features. Lastly, this latest work utilized all four classes from the BCI Competition IV-2a dataset (left and right hands, feet, and tongue-related classes), highlighting that there were no additional artifact removal methods used other than the current preprocessing filtering mentioned above.

This study introduces features from the field of emotion recognition (ER), which are specifically designed to capture complex, non-linear brain patterns, which can address some challenges in the motor imagery field, like the low signal-to-noise ratio (SNR) and difficulty in distinguishing between similar MI tasks (i.e., left-hand vs. right-hand movements).

The cross-application of both domains, ER-EEG and MI-EEG, is the novel approach that this study offers. The lack of research in applying emotion recognition (ER) feature extraction methods to motor imagery (MI) tasks justifies the need for our current research.

3. Background and Theory

This section covers the foundational elements required to understand the methodology and scope of the current study. Among these essential topics are machine learning classifiers, feature extraction methods, and preprocessing techniques. Additionally, we present the key concepts, relevant theories, and important definitions crucial for a comprehensive understanding of this research.

3.1. Notch Filtering

The initial step in preprocessing our EEG data involves applying a notch filter [23], which is commonly used to eliminate a targeted frequency from EEG signals. This specialized filter introduces a pronounced, narrow suppression, known as a "notch", at a designated frequency. By configuring this filter, we can reduce interference in the signals being studied using factors such as the sampling rate Fs, the powerline frequency $f0$, and the quality factor Q. The transfer function $H(z)$ represents the digital notch filter mathematically, as expressed in Equation (1):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1 + \sum_{n=1}^{N} a_n z^{-n}} \quad (1)$$

where the following apply

- b_m are the feedforward coefficients (numerator);
- a_n are the feedback coefficients (denominator);
- M is the order of the numerator;
- N is the order of the denominator;
- $X(z)$ is the Z-transform of the input signal;
- $Y(z)$ is the Z-transform of the output signal.

3.2. High-Pass Filtering

The next preprocessing step for filtering our EEG data involves using high-pass Butterworth filters (HPFs) [24], which allow higher-frequency components of EEG signals to pass while diminishing lower-frequency components. These filters are widely used to reduce low-frequency noise, effectively preserving target higher-frequency brain activity. HPFs are particularly useful for mitigating artifacts, correcting electrode drifts during recordings from scalp-based BCI systems and improving the quality of EEG data analyzed

in this study. The mathematical expression for a Butterworth high-pass filter of order N with a cutoff frequency ωc is generally given in Equation (2):

$$C(z) = \frac{B(z)}{A(z)} \tag{2}$$

Let $B(z)$ and $A(z)$ denote the Z-transforms of the filter coefficients b and a, respectively. A fourth-order high-pass filter is implemented, with its transfer function incorporating b_i and a_i, as given by (3):

$$C(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} \tag{3}$$

To configure the filter, we compute the coefficients b and a for a fourth-order high-pass Butterworth filter with a normalized cutoff frequency ωc. The application of this filter to EEG data requires executing multiplication in the Z-domain, as indicated in Equation (4):

$$Filtered\ Signal(z) = C(z) \cdot EEG\ Signal(z) \tag{4}$$

3.3. CAR Montage Filter

The final preprocessing method applied to our EEG data is the common average reference (CAR) montage [25]. This technique in EEG signal processing is designed to minimize noise shared across all EEG channels. By re-referencing each channel to the collective average of all channels, CAR enhances the clarity of brain electrical activity relative to this average, thereby improving the signal-to-noise ratio and the overall quality of the data. Let X denote the matrix of EEG signals, where rows represent time samples and columns represent channels. The CAR operation is mathematically expressed in Equation (5):

$$CAR = X - \frac{1}{N}\sum_{i=1}^{N} X_i \tag{5}$$

where the following apply:

- CAR is the EEG data matrix after the filter has been applied;
- X represents the original EEG data matrix;
- N denotes the total number of EEG channels;
- Xi corresponds to the signal from the i-th channel.

3.4. Statistical Features

The initial feature extraction approach involves statistical features [26], which are classified as time-domain characteristics. Widely applied in EEG signal processing, these features play a significant role in brain–computer interface applications and seizure detection. The four core measures include mean, variance, skewness, and kurtosis.

Mean ($\overline{x}$): This is the average value of the signal and is calculated using Equation (6):

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

where N is the total number of samples, and x_i represents the i-th sample in the signal.

Variance (σ^2): Variance measures how much the data points deviate from the mean, given by Equation (7):

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{7}$$

Skewness: This metric accesses the asymmetry of the probability distribution of a real-valued random variable in relation to its mean. It is calculated as shown in Equation (8):

$$Skewness_{stat} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \overline{x}}{\sigma}\right)^3 \tag{8}$$

where σ is the standard deviation of the signal.

Kurtosis: Kurtosis evaluates the presence of extreme values (outliers) in the tails of a distribution, similar to skewness, in that it considers the probability distribution of a real-valued random variable. It is determined using Equation (9):

$$Kurtosis_{stat} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \overline{x}}{\sigma}\right)^4 - 3 \quad (9)$$

where the subtraction of 3 adjusts the kurtosis so that a normal distribution has a kurtosis value of zero.

3.5. Wavelet Analysis Features

The second feature extraction technique is wavelet transform [27], classified as time–frequency analysis features. These are highly effective in EEG data analysis, as they break down signals into various frequency components while preserving temporal information. In this study, we used the discrete wavelet transform (DWT) to derive features from our EEG dataset. Our approach emphasizes calculating the mean absolute value of wavelet coefficients at each decomposition level, which reflects the average magnitude of these coefficients and serves as an indicator of the signal's energy, as shown in Equation (10):

$$\mu_{t,c}^{(j)} = \frac{1}{N_j} \sum_{k=1}^{N_j}\left|c_{t,c}^{(j)}[k]\right| \quad (10)$$

where the following apply:

- $\mu_{t,c}^{(j)}$ is the mean absolute value at level j for channel c in trial t;
- N_j is the total number of coefficients at level j;
- $\left|c_{t,c}^{(j)}[k]\right|$ denotes the absolute value of the $k-th$ coefficient at level j.

3.6. Higher-Order Spectra Features

The third feature extraction technique consists of Higher-Order Spectra (HOS) features [28], which are considered frequency-domain characteristics. These features are derived from the power spectral density (PSD), providing insights into power fluctuations within EEG signals and aiding in examining the Euclidean distribution of electrical activity across frequencies. In this analysis, skewness and kurtosis are selected as key HOS features. Skewness indicates the symmetry of the PSD around its mean, while kurtosis assesses the shape of the PSD distribution, focusing on the presence of heavy tails or potential outliers. These features are computed using Equations (11) and (12):

$$Skewness_{HOS} = \frac{\frac{1}{N}\sum_{i=1}^{N}(Pxx_i - P\overline{x}x)^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(Pxx_i - P\overline{x}x)^2}\right)^3} \quad (11)$$

$$Kurtosis_{HOS} = \frac{\frac{1}{N}\sum_{i=1}^{N}(Pxx_i - P\overline{x}x)^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(Pxx_i - P\overline{x}x)^2\right)^2} - 3 \quad (12)$$

where the following apply:

- Pxx_i represents the individual power spectral density values;
- $P\overline{x}x$ represents their mean;
- N denotes the number of observations in the PSD.

3.7. Hjorth Features

The fourth feature extraction approach involves Hjorth parameters [29,30], classified as time-domain features. These statistical metrics characterize time-domain signals through

three key aspects: activity, mobility, and complexity. Together, they offer meaningful insights into the signal's energy, frequency composition, and degree of irregularity.

Activity Parameter A: quantifies the variance of the EEG signal and is calculated using (13):

$$A = \mathrm{Var}(x(t)) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{13}$$

where the following apply:

- x_i represents the signal samples;
- $\overline{x}$ represents the mean value of the signal;
- N denotes the total number of samples.

Mobility Parameter M: measures the standard deviation of the first derivative of the EEG signal normalized by the activity. It is given by (14):

$$M = \sqrt{\frac{\mathrm{Var}(\Delta x(t))}{A}} = \sqrt{\frac{\frac{1}{N-1}\sum_{i=1}^{N-1}(\Delta x_i)^2}{A}} \tag{14}$$

where the following apply:

- $\Delta x(t)$ denotes the first derivative of $x(t)$;
- $\Delta x_i = x_{i+1} - x_i$ represents the first differences between consecutive signal samples.

Complexity Parameter C: the ratio of the mobility of the first derivative of the signal to the mobility of the signal itself, expressed as (15):

$$C = \frac{M_{\Delta x(t)}}{M} = \frac{\sqrt{\frac{\mathrm{Var}(\Delta^2 x(t))}{\mathrm{Var}(\Delta x(t))}}}{M} \tag{15}$$

where the following apply:

- $\Delta^2 x(t)$ represents the second derivative of $x(t)$;
- $\Delta^2 x_i = \Delta x_{i+1} - \Delta x_i$ represents the second differences in the signal.

3.8. Fractal Dimension Features

The fifth and final feature extraction method is based on fractal dimension features, which fall under time–frequency analysis characteristics. In this study, we applied several fractal dimension algorithms commonly utilized in EEG analysis, specifically the Katz, Petrosian, and Higuchi methods. Fractal dimensions offer insights into the self-similarity and complexity of the signal.

The Katz fractal dimension [11] assesses signal complexity by examining how the signal's structure occupies space as it scales. This method evaluates deviations from smoothness by considering both the total length of the signal path and its spatial extent, as expressed in Equation (16):

$$FD_{\mathrm{Katz}} = \frac{\ln(n)}{\ln(n) + \ln\left(\frac{d}{\mathrm{L}}\right)} \tag{16}$$

where the following apply:

- ln denotes the natural logarithm;
- n is the total number of observations (data points) in the time series;
- L is the total length of the signal path;
- d is the maximum distance from the first point to any other point in the signal.

Petrosian fractal dimension [12] measures the irregularity or self-similarity within a signal by calculating the number of sign changes in its first derivative. It is defined by Equation (17):

$$FD_{Petrosian} = \frac{\log_{10}(n)}{\log_{10}(n) + \log_{10}\left(\frac{n}{n+0.4\cdot N_\delta}\right)} \quad (17)$$

where the following apply:

- $\log_{10}$ is the base-10 logarithm;
- n is the total number of observations (data points) in the time series;
- N_δ represents the number of sign changes in the derivative, providing insights into the signal's frequency content and waveform complexity.

The Higuchi fractal dimension [13] assesses the roughness or complexity of a signal by examining its variation as it undergoes successive downsampling, represented by Equations (18)–(20):

$$HFD = -\text{slope}(\log(x), \log(L)) \quad (18)$$

$$L_k = \frac{1}{k}\sum_{m=0}^{k-1}\left(\frac{N-1}{k(N-m-1)}\sum_{i=1}^{\lfloor\frac{N-m}{k}\rfloor}|data[m+ik] - data[m+(i-1)k]|\right) \quad (19)$$

$$x = [1, 2, \ldots, k_{\max}] \quad (20)$$

where the following apply:

- HFD is the Higuchi fractal dimension resulting from the computation;
- L_k is the average length over k sets;
- k is an integer defining the time interval for calculating L_k;
- $k_{\max}$ is the maximum No. of k, defining different scales at which the time series is analyzed;
- N is the total No. of data points;
- m ranges from 0 to $k-1$;
- $L_{m,\,k}$ denotes the length of the curve for a particular scale k and starting point m;
- i indexes the sum over $N-m$ data points;
- $data$ refers to the actual EEG time series being analyzed;
- x is a list of integers from 1 to $k_{\max}$, which are used alongside L to compute the slope.

3.9. GSVM Classifier (SVM with RBF Kernel)

The initial supervised machine learning classifier used in this study is the Gaussian radial basis function (RBF) support vector machine (GSVM) [31], a variant of the traditional SVM. This approach incorporates the radial basis function, also known as the Gaussian kernel, within the support vector machine framework. It is especially well suited for handling non-linearly separable data, a common scenario in EEG signal processing. The Gaussian RBF kernel is defined in Equation (21):

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\left|\mathbf{x} - \mathbf{x}'\right|^2\right) \quad (21)$$

where the following apply:

- K is the RBF kernel function;
- $\mathbf{x}$ are feature vectors in the input space;
- $\mathbf{x}'$ are other feature vectors in the input space;
- γ is a parameter that defines how much influence a single training example has.

The decision function of SVM in this kernel-induced feature space is expressed as follows in (22):

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \quad (22)$$

where the following apply:

- N is the number of support vectors;
- α_i are Lagrange multipliers obtained by solving the SVM optimization problem;
- y_i are the class labels of the training examples;
- b is the bias term.

3.10. CART Classifier

The second supervised machine learning classifier applied here is the classification and regression tree (CART) [32], which uses decision trees for both classification and regression tasks. In EEG signal analysis, CART is often employed to classify various mental states or detect patterns related to specific neurological conditions. The algorithm can be outlined using the following pseudo-code:

Given a set of EEG features X and a target variable Y, the following procedure applies:

1. Start with the root node containing all instances.
2. If all instances have the same value for Y, stop. Otherwise, proceed.
3. Select feature x_i and threshold θ to split:

$$x_i, \theta = \underset{x,t}{\text{argmin}} \text{Impurity}(X, Y, x, t)$$

4. Split the node into two child nodes:

$$X_{\text{left}} = \{x \in X | x_i \leq \theta\}$$

$$X_{\text{right}} = \{x \in X | x_i > \theta\}$$

5. Repeat steps 2–4 recursively for X_{left} and X_{right}.
6. Stop if a maximum tree depth is reached or if further splitting does not improve impurity measures significantly.

where the following apply:

- X is the set of input variables (EEG features);
- Y is the target variable (e.g., type of brain activity);
- x_i is a feature from the EEG features;
- θ is the threshold value used for splitting a node;
- $\text{Impurity}(X, Y, x, t)$ is a measure of the homogeneity of the target variable within the nodes after the split (it can be Gini impurity, entropy, or another suitable metric);
- X_{left} is a subset of the data where the value of x_i is less than or equal to θ;
- X_{right} is a subset of the data where the value of x_i is greater than θ;
- $\underset{x,t}{\text{argmin}}$ is the argument of the minimum, the values of x and t that minimize the impurity.

3.11. Linear SVM Classifier

The third supervised machine learning classifier being used is the linear support vector machine (SVM) [33]. In EEG, it aims to find the hyperplane that maximizes the margin between two classes. The margin is the distance between the hyperplane and the nearest data points from each class, known as support vectors. The decision function of a LinearSVM is defined as (23):

$$f(x) = \boldsymbol{w}^\top \boldsymbol{x} + b \tag{23}$$

where the following apply:

- $\boldsymbol{x}$ is the input feature vector derived from EEG signals;
- $\boldsymbol{w}$ is the weight vector perpendicular to the hyperplane;
- b is the bias term.

3.12. SVM with Polynomial Kernel Classifier

The fourth supervised machine learning classifier is the support vector machine (SVM) with a polynomial kernel [33]. In EEG, it is beneficial when the relationship between input features and target classes is non-linear. The polynomial kernel allows the SVM to project the input data into a higher-dimensional feature space where a linear separator can be found. This kernel function is defined as (24):

$$K(\boldsymbol{x}, \boldsymbol{x}') = \left(\gamma \boldsymbol{x}^{\top} \boldsymbol{x}' + r\right)^{d} \tag{24}$$

where the following apply:

- $K(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel function measuring similarity between two feature vectors;
- $\boldsymbol{x}$ & $\boldsymbol{x}'$ are input feature vectors derived from EEG signals;
- γ is a scaling factor that adjusts the influence of the features (kernel coefficient);
- r is a constant that controls the trade-off between higher-order and lower-order terms;
- d is the degree of the polynomial, determining the complexity of the decision boundary.

The decision function for the SVM with a polynomial kernel is expressed as (25):

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{25}$$

where the following apply:

- N is the number of support vectors selected during training;
- α_i are Lagrange multipliers obtained by solving the SVM optimization problem;
- y_i are the class labels of the training samples;
- $\boldsymbol{x_i}$ are the support vectors from the training data;
- b is the bias term or intercept.

3.13. Cross-Validation Metrics

The cross-validation accuracy was used to evaluate the model's performance [34]. This method helps determine how well a machine learning model is likely to perform. The process involves splitting the dataset into k parts or folds. For example, let us say we have a dataset D, divided into k folds, with each fold labeled as D_i, where i ranges from 1 to k. For each fold, the model is trained on the rest of the dataset, excluding D_i, and then tested on D_i to calculate a performance score, S_i. The overall performance is then determined by averaging the scores across all folds, as shown in the formula below (26):

$$CV = \frac{1}{k} \sum_{i=1}^{k} S_i \tag{26}$$

where the following apply:

- CV is the cross-validation score;
- k is the number of folds;
- S_i is the performance score obtained when using the fold D_i as the test set and the remaining $k-1$ fold as the training set (S_i can be accuracy, which is used in this study, precision, recall, F1 score, mean squared, etc.).

4. Materials and Methods

This section describes the evaluation strategy adopted in the research and outlines the experimental setup used for testing the system. Additionally, it explains the implementation of the proposed pipeline and provides a detailed description of the architectural design. Moreover, an overview of the dataset utilized in the study is presented to offer context for the subsequent analysis. Lastly, this section outlines the specific parameters that were fine-tuned during the study, explaining how these adjustments contributed to the overall performance and outcomes.

4.1. EEG Dataset

The dataset considered has an original general data format for biomedical signals (GDF); those files contain the training and evaluation data for each subject in question. However, we use the latter's .mat (MATLAB R2022a files) equivalents. The dataset name is the BCI Competition IV 2a dataset [15]. Each subject's data contain several event types, with the most significant being the start of a trial and the cues for different motor imagery tasks: left hand (class 1), right hand (class 2), foot (class 3), and tongue (class 4). Each subject and session are represented by a single MAT file. The motor imagery tasks follow a specific structure, as illustrated in Figure 1.

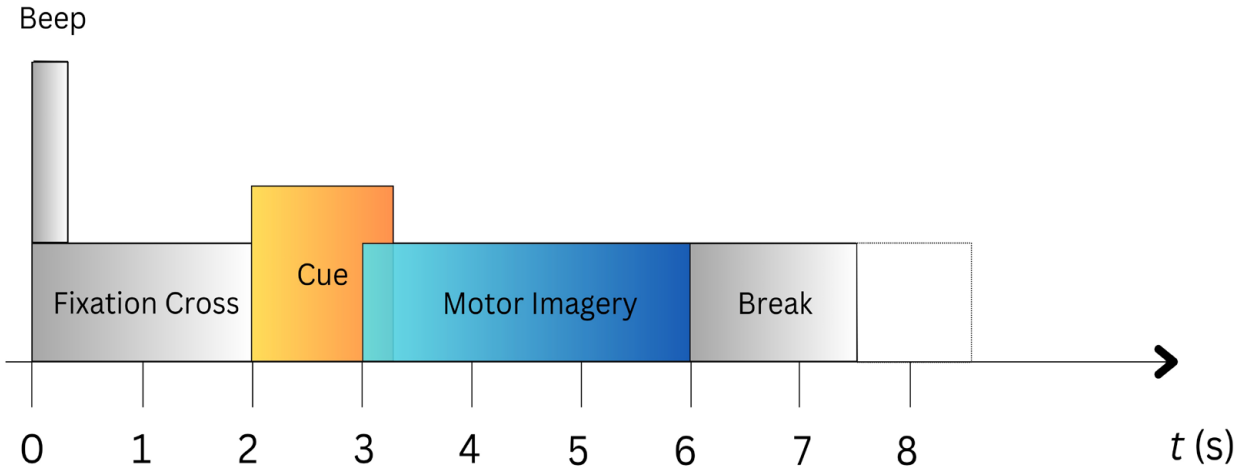

Figure 1. The timing diagram for the motor imagery tasks.

Each session begins with a sequence of runs, starting with the first run of 2 min with eyes open, then a second run of 1 min with eyes closed, and concluding with a third run of 1 min involving eye movements. Across the two sessions, each contains six runs comprising 288 trials, with durations ranging from 2016 to 2304 s. On average, each trial lasts between 7 and 8 s, with nine runs in total, each consisting of 48 trials, resulting in durations between 336 and 384 s per run.

Given that the first 22 channels are allocated for EEG recordings, we excluded the final 3 channels, which were originally designed for monopolar electrooculography (EOG) applications in visual research studies. The EEG data, recorded using 25 channels at a sampling rate of 250 Hz, underwent a bandpass filter between 0.5 Hz and 100 Hz, with an additional 50 Hz notch filter applied to eliminate line noise.

4.2. Embraced Pipeline

The adopted pipeline integrates both ER-EEG and MI-EEG, as illustrated in Figure 2. This study utilizes 24 Jupyter Notebooks adopting six distinct feature sets and four classifiers. Each classifier–feature set combination is applied across nine subjects, with four classes (left hand, right hand, foot, and tongue) considered for analysis. The classifiers implemented are SVM with rbf kernel (GSVM), CART regression tree, LinearSVM, and SVM with a polynomial kernel. These were taken from [7,14].

This section introduces a detailed justification of the preprocessing steps, the feature extraction methods' relevance, and the justification for classifier selection.

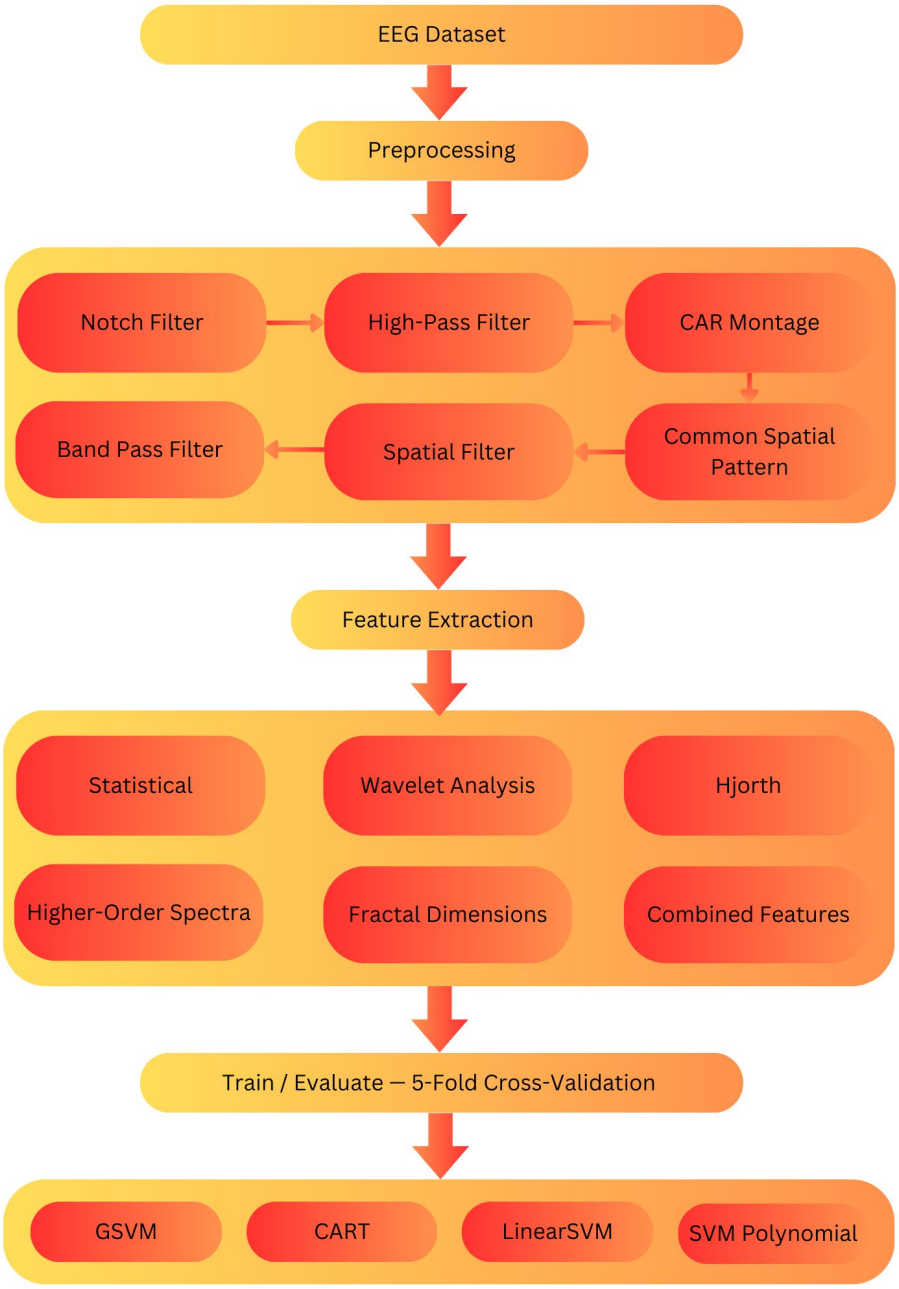

Figure 2. The embraced ER/MI hybrid pipeline.

The preprocessing pipeline begins with a notch filter [23] to remove powerline noise, which is a prevalent artifact in EEG signals. Removing this interference early in the sequence prevents the latter from affecting subsequent filtering and spatial transformations. A high-pass filter [24] follows the notch filter to remove the low-frequency drifts from the signal, which helps stabilize the baseline by removing unneeded fluctuations. To reduce common noise, a CAR montage [25] is applied to improve the signal-to-noise ratio factor. We included the initial reference channel in the CAR computation, ensuring the preservation of the data's full rank [35]. To verify this, the calculation of the rank of the EEG data before and after applying the CAR montage was applied, and we found that the rank remained unchanged. The next preprocessing stage is the implementation of the common spatial pattern (CSP) [36] to identify the filters that maximize variance between different motor imagery tasks. Following CSP, spatial filtering [37] is applied to highlight any task-relevant regions in the EEG signals. Lastly, a band-pass filter [38] is applied at the end of the preprocessing pipeline to focus on the main specific rhythms.

Regarding the relevance of feature extraction techniques, the statistical features [26] capture basic distribution properties of the EEG signal, such as mean, variance, skewness, and kurtosis. In MI tasks, the latter provide information about the amplitude changes and its variability. Wavelet analysis [27] provides a time–frequency decomposition of the EEG signal, enabling the detection of momentary task-related patterns. Higher-order spectra (HOS) [28] features capture non-linear interactions in the EEG signal by analyzing higher-order statistics like skewness and kurtosis of the power spectral density. Hjorth [29,30] parameters, such as activity, mobility, and complexity, summarize the EEG signal's structure, capturing both amplitude and frequency characteristics. Fractal dimensions (Katz, Higuchi, Petrosian) [11–13] measure the EEG signal's complexity and self-similarity, which can benefit MI tasks by capturing the varying structural complexity inherent in motor imagery. Lastly, the combined feature set integrates statistical, wavelet, Hjorth, HOS, and fractal dimension features to offer a full representation of the EEG signals. This combination introduces noise but still increases the chances of capturing meaningful patterns across different aspects of the EEG signals in the study.

The four classifiers used, LinearSVM, CART, GSVM, and SVM with a polynomial kernel, were selected to provide a detailed comparison of both linear and non-linear approaches to EEG classification. LinearSVM was chosen because it is well-known for handling high-dimensional data effectively, as supported by EEG studies [33]. CART, a decision tree method, provides flexible decision making for non-linear patterns but can sometimes be overfitted with highly dimensional EEG features [32]. GSVM [31] and polynomial SVM [33] build on SVM's strengths by capturing non-linear relationships, making them useful for modeling complex connections between EEG channels.

4.3. Experimental Setup

The development environment this study adopted relied on several essential python packages, including numpy for managing EEG signals and multidimensional arrays, scipy for applying high-level mathematical functions to these arrays, pandas for handling data in the form of data frames, matplotlib for creating advanced line plots, seaborn for creating statistical visualizations like heatmaps, and sklearn for implementing various supervised machine learning algorithms. To manage the environment and streamline the workflow for testing and training procedures, we utilized Conda. This package management comes in both a graphical user interface (GUI), named Anaconda, and a command line tool (CLI), named mini-forge Conda. During our experimentation, the CLI version was used. This system facilitated the installation, execution, and updating of necessary packages and their dependencies, ensuring smooth operation throughout the study. The interactive coding environment was provided by Jupyter Notebook and Jupyter Lab, which supported Python and allowed for the inclusion of rich text elements such as Markdown and LaTeX and dividing the Python codebase into cells, which enhanced the development of the workflow. As everything is running locally, the speed of code execution is noticeable. This interactive environment proved vital in developing and testing our code dynamically and flexibly. The code was written in Python 3.9.16 and developed using Visual Studio Code 1.93.1 (Universal), a widely used integrated development environment (IDE). Visual Studio Code offers features such as local build automation (instead of the use of notebooks on the Google Collab servers), version control system integration (used for rolling back or switching code branches), and debugging tools (coding line-by-line interpretation), which enhanced the efficiency of the development process. All these tasks were performed on a MacBook Air powered by the M1 Apple Silicon Chip with 16 GB of RAM (Apple Inc., Cupertino, CA, USA), which provided sufficient computational power for our custom pipeline implementation.

4.4. Evaluation Strategy

To ensure our model's robustness, we implemented a k-fold cross-validation [34] strategy with k = 5. Cross-validation is especially valuable in EEG-based studies because

of the high variability between subjects and sessions, which is the case regarding our BCI IV 2a dataset. This strategy allows us to assess different classifiers' performance across different data segments, targeting a minimization of the risk of overfitting and offering a more consistent estimate of each model's accuracy, whether it is GSVM, CART, LinearSVM, or SVM with a polynomial kernel.

As our hybrid pipeline utilizes k-fold cross-validation, it provides robust accuracy estimates across multiple data segments. Specifically, we apply 5-fold cross-validation, which ensures that approximately 80% of the data is used for training and 20% for evaluation per fold. This split allows each data point to be part of the evaluation set on one side and the training set on the other; this will reduce the risk of overfitting while taking into account a balanced generalization approach.

4.5. Tweaked Parameters

The parameters used in this research were tweaked to extract the most out of each use case experiment. The parameter overview is presented in Table 1.

Table 1. The list of tweaked parameters.

Parameter	Related to	Value
notch_filter_fs	Notch Filter	250.0
notch_filter_powerline_freq	Notch Filter	50
notch_filter_q	Notch Filter	30.0
high_pass_filter_fs	High-Pass Butterworth Filter	250
high_pass_cutoff	High-Pass Butterworth Filter	0.5
car_montage_filter_axis	CAR Montage Filter	1
car_montage_filter_keep_dims	CAR Montage Filter	True
wavelet_value	Wavelet Analysis	db4
wavelet_level	Wavelet Analysis	5
hos_fs	Higher-Order Spectra	250.0
hos_nperseg	Higher-Order Spectra	256
hos_noverlap	Higher-Order Spectra	128
fd_higuchi_k_max_value	Fractal Dimension: Higuchi	10
filter_bank_type	Filters Bank with Butterworth Filter	butter
filter_bank_order	Filters Bank with Butterworth Filter	2
filter_bank_max_freq	Filters Bank with Butterworth Filter	40
time_windows_flt	Filters Bank with Butterworth Filter	[2.5, 3.5]...[2.5, 6]
bw	Bandwidth of Filtered Signals	[2, 4, 8, 16, 32]
no_csp	Number of CSP Features	24
cart_max_depth	CART Classifier	10
cart_random_state	CART Classifier	1
cart_criterion	CART Classifier	gini
cart_splitter	CART Classifier	best
cart_min_samples_split	CART Classifier	2
cart_min_samples_leaf	CART Classifier	1
linear_svc_c	LinearSVM Classifier	0.1
linear_svc_intercept_scaling	LinearSVM Classifier	1
linear_svc_loss	LinearSVM Classifier	hinge
linear_svc_max_iter	LinearSVM Classifier	1000
linear_multi_class	LinearSVM Classifier	ovr
linear_svc_penalty	LinearSVM Classifier	l2
linear_svc_random_state	LinearSVM Classifier	1
linear_svc_tol	LinearSVM Classifier	0.00001
svc_w_poly_kernel_c	SVM with Polynomial Kernel Classifier	0.1
svc_w_poly_kernel_type	SVM with Polynomial Kernel Classifier	poly

Table 1. *Cont.*

Parameter	Related to	Value
svc_w_poly_kernel_degree	SVM with Polynomial Kernel Classifier	10
svc_w_poly_kernel_gamma	SVM with Polynomial Kernel Classifier	auto
svc_w_poly_kernel_coef0	SVM with Polynomial Kernel Classifier	0.0
svc_w_poly_kernel_tol	SVM with Polynomial Kernel Classifier	0.001
svc_w_poly_kernel_cache_size	SVM with Polynomial Kernel Classifier	10000
svc_w_poly_kernel_max_iter	SVM with Polynomial Kernel Classifier	−1
svc_w_poly_kernel_decision_fx	SVM with Polynomial Kernel Classifier	ovr
gsvm_c	SVM with RBF Kernel (GSVM) Classifier	20
gsvm_kernel_type	SVM with RBF Kernel (GSVM) Classifier	rbf
gsvm_degree	SVM with RBF Kernel (GSVM) Classifier	10
gsvm_gamma	SVM with RBF Kernel (GSVM) Classifier	auto
gsvm_coef0	SVM with RBF Kernel (GSVM) Classifier	0.0
gsvm_tol	SVM with RBF Kernel (GSVM) Classifier	0.001
gsvm_cache_size	SVM with RBF Kernel (GSVM) Classifier	10000
gsvm_max_iter	SVM with RBF Kernel (GSVM) Classifier	−1
gsvm_decision_fx	SVM with RBF Kernel (GSVM) Classifier	ovr
fs	Sampling Frequency	250.0
no_channels	Number of EEG Channels	22
no_subjects	Number of Subjects	9
no_classes	Number of Classes	4
no_splits	Number of Folds in Cross-Validation	5

The fine-tuning process required many attempts to find the most suitable parameters for this study. The approach started with optimizing the preprocessing filters, feature sets, and the performance of the classifiers used.

The first step was preprocessing filters from the ER pipeline. The Notch filtering used a powerline frequency, $f0$, of 50 Hz, which describes the center frequency to be removed, and a quality factor value, Q, of 30, which controls the bandwidth of the notch. Second, the high-pass filter used a cutoff frequency, ωc, of 0.5 Hz, the lowest frequency retained. Third, the CAR montage used a filtered axis value of 1 and kept the dimension parameter set to true. This ensures that the CAR montage is correctly applied across channels and preserves the data structure.

This was followed by the preprocessing filters from the MI pipeline. The total number of CSP filters across all classes and time windows is 24. The filter bank type is *butter*, the adopted order is 2, and the maximum frequency is 40 Hz. The bandwidth of filtered signals, *bw*, are grouped in the following list [2,4,8,16,32] in Hz. The latter provides band-specific filtering for the EEG signal, capturing distinct frequency bands like alpha and beta. Lastly, the time windows (start and end time) are set and then scaled by the current sampling frequency, 250 Hz, which allows the CSP filters to operate over multiple temporal windows, capturing time-specific features across different parts of the EEG signal.

Next, the optimization of each feature set with a parameter to be fine-tuned is disclosed, which includes wavelet analysis, higher-order spectra, and the fractal dimension's Higuchi. First, in wavelet analysis, the selected wavelet type to be used in the decomposition is *db4,* and the selected decomposition level is 5. Second, for higher-order spectra, the selected number of segments for calculating the power spectral density (PSD), *nperseg*, is 256, and the number of overlaps between segments, *noverlap*, is 128. Third, for fractal dimension's Higuchi, the maximum number of intervals, *kmax*, is 10, which defines the maximum scaling factor for constructing the needed subsequences.

Finally, the performance of the classifiers used, GSVM, CART, LinearSVM, and SVM with a polynomial kernel, is tweaked, and 5-fold cross-validation is applied. First, the GSVM classifier has a C value of 20, an *rbf* kernel type, a degree of 10, an *auto* gamma value, a *coef0* of 0.0, a *tol* of 0.001, a cache size of 10000, a max iteration of −1, and a decision function of *ovr*. Second, the CART classifier has a max depth of 10, a random state of 1,

and a criterion of *gini*. A *best* splitter is considered, the minimum sample split is 2, and the minimum sample leaf is 1. Third, LinearSVM has a *C* value of 0.1, a scaling interception value of 1, a *hinge* loss, a maximum iteration of 1000, an *ovr* multi-classification, a penalty of *l2*, a random state of 1, and a *tol* of 0.00001. Lastly, LinearSVM with a polynomial kernel, a *C* value of 0.1, a *poly* kernel type, a kernel degree of 10, an *auto* gamma, a *coef0* of 0.0, a tol of 0.001, a cache size of 10000, a maximum iteration of −1, and an *ovr* decision function.

5. Results

In this section, we present the results obtained in detail, along with a thorough discussion of their significance.

5.1. Statistical Features

The results of the statistical method indicate 11,440 total features. Subject No. 1 achieved the highest accuracy with LinearSVM at 83.12%; however, GSVM's accuracy was moderate, yielding 71.77%. Subject No. 2 had a poor overall score across all classifiers, with the highest being LinearSVM, at 68.14%. Subject No. 3 showed excellent performance with LinearSVM, at 89.62%; GSVM had a reduced accuracy in comparison: 52.59%. Subject No. 7, on the other hand, showed consistently high performance across all classifiers: 90.40% for LinearSVM and 85.64% for CART.

We deduce that the lowest average success rate is related to SVM with a polynomial kernel (52.25%). GSVM is slightly better (55.68%), but CART outperforms it (59.30%), and LinearSVM has the highest accuracy (77.10). The results of applying statistical methods for feature extraction are presented in Table 2.

Table 2. The results of the statistical features.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	83.12	66.01	71.77	67.01
Subject No. 2	68.14	41.48	49.62	41.48
Subject No. 3	89.62	72.59	52.59	56.66
Subject No. 4	63.70	38.92	53.70	48.07
Subject No. 5	77.89	54.20	66.05	48.13
Subject No. 6	47.85	32.83	31.05	33.79
Subject No. 7	90.40	85.64	69.38	61.62
Subject No. 8	86.70	64.41	54.16	59.11
Subject No. 9	86.46	77.62	52.76	54.40
Avg. Success Rate	77.10	59.30	55.68	52.25

5.2. Wavelet Analysis Features

The results of the wavelet analysis method indicate 11,484 total features. Subject No. 1 performed remarkably well across all classifiers, with the highest being 83.48% for general LinearSVM. Subject No. 2 showed lower performance across all classifiers, with LinearSVM being the most optimal, at 68.51%. Subject No. 3 attained excellent results, with LinearSVM reaching 89.62% and GSVM slightly close behind (85.55%). Subject No. 7 also showed high performance, with LinearSVM at 90.03%. Subject No. 6 still showed low accuracy, with LinearSVM reaching 50.16% and CART being the lowest, at 31.93%.

Regarding the classifiers' overall average success rate, the lowest classification was attributed to CART (58.95%). SVM with a polynomial kernel is better, reaching 71.44% accuracy; GSVM scores reach 75.25%; and the highest success rate is reached with LinearSVM (77.25%). The results of applying wavelet analysis methods for feature extraction are presented in Table 3.

Table 3. The results of wavelet analysis features.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	83.48	65.60	83.13	80.18
Subject No. 2	68.51	45.55	63.70	60.37
Subject No. 3	89.62	72.96	85.55	84.81
Subject No. 4	62.17	35.89	64.42	60.61
Subject No. 5	77.48	59.57	73.28	73.30
Subject No. 6	50.16	31.93	47.89	41.97
Subject No. 7	90.03	81.21	88.56	80.45
Subject No. 8	88.59	60.64	85.92	84.05
Subject No. 9	85.16	77.18	84.77	77.20
Avg. Success Rate	77.25	58.95	75.25	71.44

5.3. Higher-Order Spectra Features

The results of the higher-order spectra method indicate 11,396 total features. Subject No. 1 performed well, with GSVM at 83.15% and LinearSVM at 82.03%. However, SVM with a polynomial kernel and CART, respectively, at 74.35% and 63.79%, were less effective. Subject No. 2 showed a moderate outcome, with LinearSVM at 68.51% accuracy. Subject No. 3 showed strong performance, with LinearSVM at 88.88%, whilst CART was equally good among other tree-based classifiers, reaching 74.44%. Subject No. 7 had the highest rate for LinearSVM, 89.66%. GSVM was also good, reaching 83.40%. Subject No. 6 had the lowest performance, with the best being LinearSVM, at 46.51%, and the rest of the classifiers performed similarly poorly. Subject No. 9 performed well across classifiers, with LinearSVM reaching 85.21% and GSVM reaching 85.18%, which shows strong results.

The lowest overall average success rate was attributed to CART, at 58.97%. SVM with a polynomial kernel had better accuracy, 63.41%, followed by GSVM at 72.71%. Still, LinearSVM outperformed all the latter, at 77.20%. The results of applying higher-order spectra methods for feature extraction are presented in Table 4.

Table 4. The results of higher-order spectra features.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	82.03	63.79	83.15	74.35
Subject No. 2	68.51	44.07	66.66	59.25
Subject No. 3	88.88	74.44	80.74	72.59
Subject No. 4	64.07	34.73	55.65	44.97
Subject No. 5	80.54	57.27	71.37	57.27
Subject No. 6	46.51	33.78	43.76	35.60
Subject No. 7	89.66	83.44	83.40	68.25
Subject No. 8	89.36	62.48	84.44	79.90
Subject No. 9	85.21	76.74	85.18	78.48
Avg. Success Rate	77.20	58.97	72.71	63.41

5.4. Hjorth Features

The results of the Hjorth method indicate 11,418 total features. Subject No. 1 showed good performance across classifiers, with LinearSVM reaching 83.11% accuracy, and CART and GSVM lagging behind. Subject No. 2 performed moderately well with LinearSVM, reaching 68.88%, but GSVM, CART, and SVM with a polynomial kernel showed a lower mark, around 50% or less for each. Subject No. 3 had strong scores, particularly LinearSVM (88.14%), although GSVM performed poorly (52.96%). For Subject No. 7 with LinearSVM, 90.02% was reached. Subject No. 6 showed the lowest overall performance, with LinearSVM at 48.81%, and other classifiers performed poorly, the same as the latter. Subject No. 9 performs well when using LinearSVM, reaching 84.76%; however, GSVM performed worse (53.18%).

The lowest average success classifier score was SVM with a polynomial kernel (52.26%). GSVM and CART were slightly better (55.90% and 59.00%, respectively), yet LinearSVM still dominated and had the highest score (77.19%). The results of applying Hjorth methods for feature extraction are presented in Table 5.

Table 5. The results of Hjorth features.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	83.11	64.90	71.40	66.63
Subject No. 2	68.88	45.55	50.37	41.48
Subject No. 3	88.14	74.44	52.96	57.03
Subject No. 4	64.86	38.14	53.70	48.07
Subject No. 5	78.66	53.05	65.68	48.12
Subject No. 6	48.81	31.03	31.51	34.24
Subject No. 7	90.02	84.16	69.75	60.88
Subject No. 8	87.44	62.52	54.54	59.49
Subject No. 9	84.76	77.20	53.18	54.41
Avg. Success Rate	77.19	59.00	55.90	52.26

5.5. Fractal Dimension Features

The results of the fractal dimensions (Katz, Petrosian, and Higuchi) method indicate 11,418 total features. Subject No. 1 performed well across all classifiers, with LinearSVM at 84.96% and GSVM at 84.24%. SVM with a polynomial kernel reached 83.50%, which is also considered vital. Subject No. 2's results are moderate: LinearSVM and GSVM both achieved 69.62%, while the CART score lagged behind at 45.55%. Subject No. 3 presented strong results, with LinearSVM and GSVM both achieving an accuracy of 90.37%. Subject No. 7 also showed strong results, with LinearSVM reaching 92.96% accuracy and GSVM reaching 91.48%. Subject No. 6 demonstrated the lowest overall performance, with all classifiers scoring around 50%, and SVM with a polynomial kernel was the highest (50.62%). Subject No. 9 performed well, with LinearSVM and GSVM reaching, respectively, scores of 87.31% and 86.87%.

The lowest classification average success rate was found with CART, 59.04%. Next was SVM with a polynomial kernel (75.31%), and GSVM was slightly better (78.11%), and the highest score was obtained with LinearSVM (79.04%). The results of applying fractal dimension methods for feature extraction are presented in Table 6.

Table 6. The results of fractal dimension features.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	84.96	64.90	84.24	83.50
Subject No. 2	69.62	45.55	69.62	62.59
Subject No. 3	90.37	74.44	90.37	85.18
Subject No. 4	65.26	38.52	63.32	56.04
Subject No. 5	82.07	53.05	79.41	79.38
Subject No. 6	47.88	31.03	47.89	50.62
Subject No. 7	92.96	84.16	91.48	86.70
Subject No. 8	90.88	62.52	89.75	88.99
Subject No. 9	87.31	77.20	86.87	84.76
Avg. Success Rate	79.04	59.04	78.11	75.31

5.6. Combined All Features

The results of combining all the previous feature sets (statistical, wavelet analysis, higher-order spectra, Hjorth, and fractal dimensions) indicate 11,748 total features. Subject No. 1 performed the best, with LinearSVM reaching 82.05%, yet GSVM performed poorly compared to the latter, reaching 53.47%. Subject No. 2 achieved moderate results. LinearSVM reached 64.81%, yet the lowest accuracy was reached by GSVM (27.40%). Subject

No. 3 achieved strong results, with LinearSVM reaching 84.07%; however, other classifiers, including GSVM, struggled, with a score of 29.25%. Subject No. 7 achieved the highest overall accuracy with LinearSVM (85.23%), and solid performance with CART (84.54%), while GSVM performed poorly, reaching 49.46%. Subject No. 6 had the lowest success rate overall, its best being LinearSVM (42.86%), and the rest of the classifiers performed poorly. Subject No. 9 achieved relatively good results, with LinearSVM reaching 77.57% accuracy.

In terms of classifiers, the lowest overall average accuracy was reached by GSVM, at 39.44%. SVM with a polynomial kernel had better accuracy (46.41%), followed by the CART, reaching 59.13%. However, LinearSVM still shines and has the best score, with a 72.94% average success rate. The results of applying all combined feature methods in feature extraction are presented in Table 7.

Table 7. The results of the combination of all feature sets.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Subject No. 1	82.05	66.35	53.47	60.08
Subject No. 2	64.81	45.18	27.40	40.00
Subject No. 3	84.07	74.07	29.25	46.29
Subject No. 4	62.92	40.07	41.53	40.39
Subject No. 5	73.68	59.93	43.15	41.64
Subject No. 6	42.86	27.36	27.38	33.76
Subject No. 7	85.23	84.54	49.46	50.92
Subject No. 8	83.28	62.14	43.57	55.31
Subject No. 9	77.57	72.57	39.69	49.32
Avg. Success Rate	72.94	59.13	39.44	46.41

5.7. Overall MI + ER Features Comparison

By comparing the global classifiers and feature sets' performances (see Table 8 and Figure 3), we can deduce that LinearSVM performs the best overall, reaching an average rate of 76.78%, achieving high accuracy across all feature sets. GSVM performs moderately well, offering an average success rate of 62.85%, which is better when employed on wavelet and fractal dimension features, but still struggles with statistical, Hjorth, and all combined features. SVM with a polynomial kernel has an average success rate of 60.18%. The best performance is achieved with fractal dimensions and wavelet analysis, but they show weaker results on other feature sets. CART consistently underperforms compared to the other SVM-based classifiers, with an average success rate of 59.07%, resulting in modest variations across all six feature sets.

Table 8. A comparison of the results obtained considering the embraced hybrid pipeline.

Classifier	LinearSVM	CART	GSVM	SVM (Kernel = "Poly")
Statistical	77.10	**59.30**	55.68	52.25
Wavelet Analysis	77.25	58.95	75.25	71.44
Higher-Order Spectra	77.20	58.97	72.71	63.41
Hjorth	77.19	59.00	55.90	52.26
Fractal Dimensions	**79.04**	59.04	**78.11**	**75.31**
Combined All	72.94	59.13	39.44	46.41
Avg. Success Rate	76.78	59.07	62.85	60.18

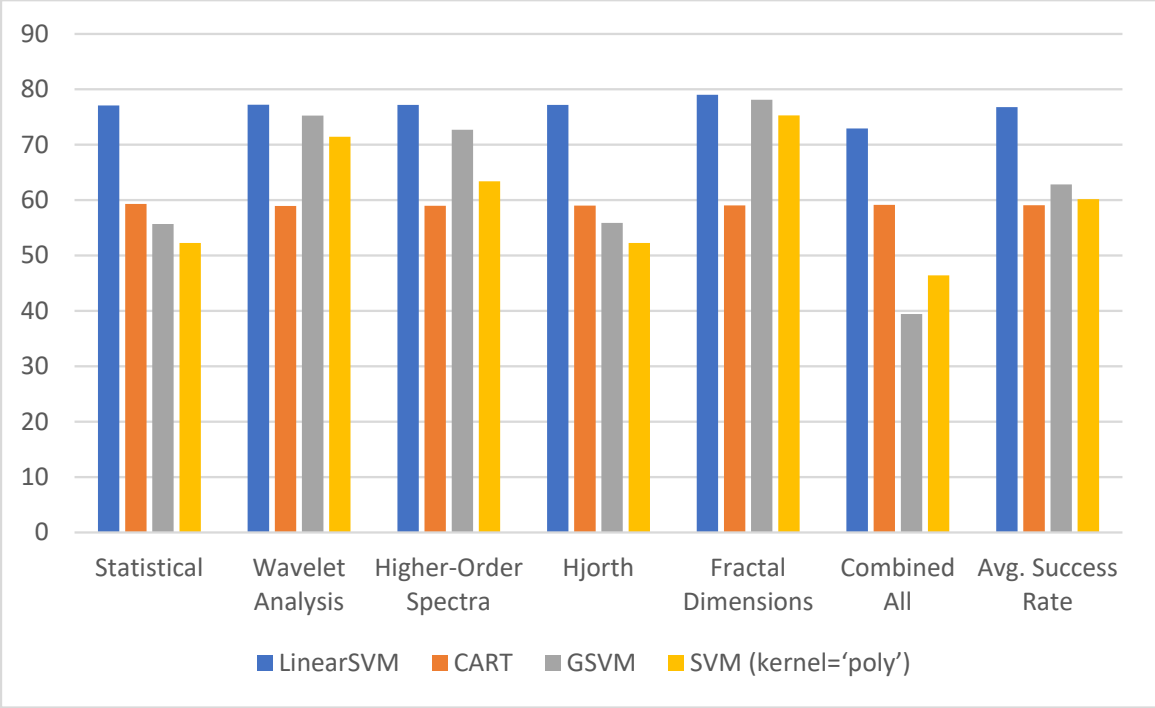

Figure 3. Clustered column chart visualizing the results generated from the hybrid ER/MI pipeline.

LinearSVM shows the most optimal results for statistical features (77.10%). Wavelet analysis's best classifier is LinearSVM (77.25%). Higher-order spectra have promising results when using LinearSVM (77.20%). Hjorth's best accuracy comes from LinearSVM (77.19%). Fractal dimensions, remarkably, provide the highest success rate compared to all other feature sets, reaching 79.04%. LinearSVM performs similarly well as using GSVM and SVM with a polynomial kernel. All combined exhibited a performance decline across all four classifiers, with LinearSVM being the most reliable. This indicates that noise is generated when combining all feature sets, resulting in a drop in the success rate.

Student's t-tests were conducted to compare the fractal dimensions feature set with other feature sets (statistical, wavelet analysis, higher-order spectra, Hjorth, and all combined). No statistically significant differences were found between fractal dimensions and any other feature set (all p-values > 0.05), indicating that fractal dimensions did not show a significant performance advantage over the other feature sets (see Table 9).

Table 9. Student's t-test findings for the fractal dimensions versus other feature sets.

Comparison	t-Statistic	p-Value
Fractal Dimensions vs. Statistical	1.8598	0.1599
Fractal Dimensions vs. Wavelet	2.6638	0.0761
Fractal Dimensions vs. HOS	1.8382	0.1633
Fractal Dimensions vs. Hjorth	1.8790	0.1568
Fractal Dimensions vs. Combined	2.0008	0.1392

The lack of significant differences between fractal dimensions and other feature sets suggests that while fractal dimensions may offer unique features, they do not outperform other sets statistically in terms of classification accuracy. This finding highlights that while fractal dimensions capture complex data characteristics, other feature sets provide similarly effective information.

5.8. Comparison with Other Published Methods

This section presents a detailed comparison between the proposed method and other previously published approaches in the motor imagery field. This comparison is based on the classifiers' mean accuracy. Comparisons include SincNet [39], HSS-ELM [40], IFNet [41], TSLDA [42], TSFBCSP-GA [43], FBRTS [44], TWSB [45], multi-scale CSP and Riemannian [14], and our novel hybrid ER/MI three-dimensional fractal dimension (Katz, Petrosian, and Higuchi) pipeline. The results for each subject related to multi-scale CSP and multi-scale Riemannian, as well as the results achieved from this current study, 3D-FD, are rounded to the nearest tenth to fit into the comparison table, Table 10. The results per year are presented in Figures 4 and 5, along with the other methods used in different publications.

Table 10. The classification accuracy of our method compared to other methods on dataset 2a.

Method	Year	A01	A02	A03	A04	A05	A06	A07	A08	A09	Mean
TSLDA	2012	80.5	51.3	87.5	59.3	45.0	55.3	82.1	84.8	86.1	70.2
Multi-Scale CSP	2018	86.8	57.2	86.5	61.4	61.2	50.7	92.4	87.8	79.1	73.7
Multi-Scale Riemannian	2018	90.0	55.4	81.3	71.9	69.6	56.7	85.6	83.8	84.9	75.5
HSS-ELM	2019	81.1	49.9	78.0	63.3	44.0	49.4	81.1	81.5	81.4	67.8
SincNet	2021	75.2	39.5	79.4	49.1	62.7	39.3	64.4	74.9	64.1	63.1
FBRTS	2022	86.1	65.2	90.0	63.8	75.6	52.4	91.1	89.0	86.5	77.7
TSFBCSP-GA	2023	86.5	59.0	89.2	69.4	63.2	54.5	87.2	80.2	81.6	74.5
IFNet	2023	88.5	56.4	91.8	73.8	69.7	60.4	89.2	85.4	88.7	78.2
TWSB	2024	89.3	66.9	89.3	69.3	74.1	60.1	89.4	88.0	85.6	**79.1**
3D-FD (Katz, Petrosian, and Higuchi)		85.0	69.6	90.4	65.3	82.1	47.9	93.0	90.9	87.3	**79.1**

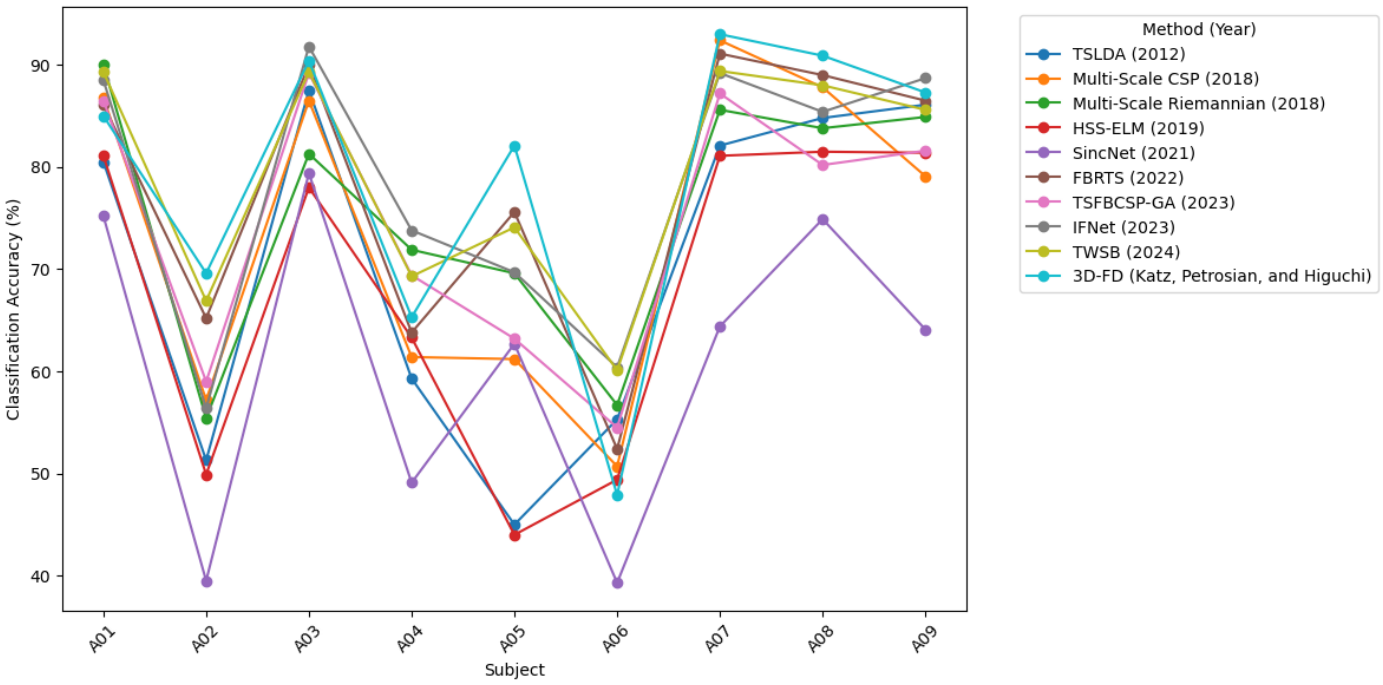

Figure 4. A line plot visualizing the classification accuracy of our method compared to others.

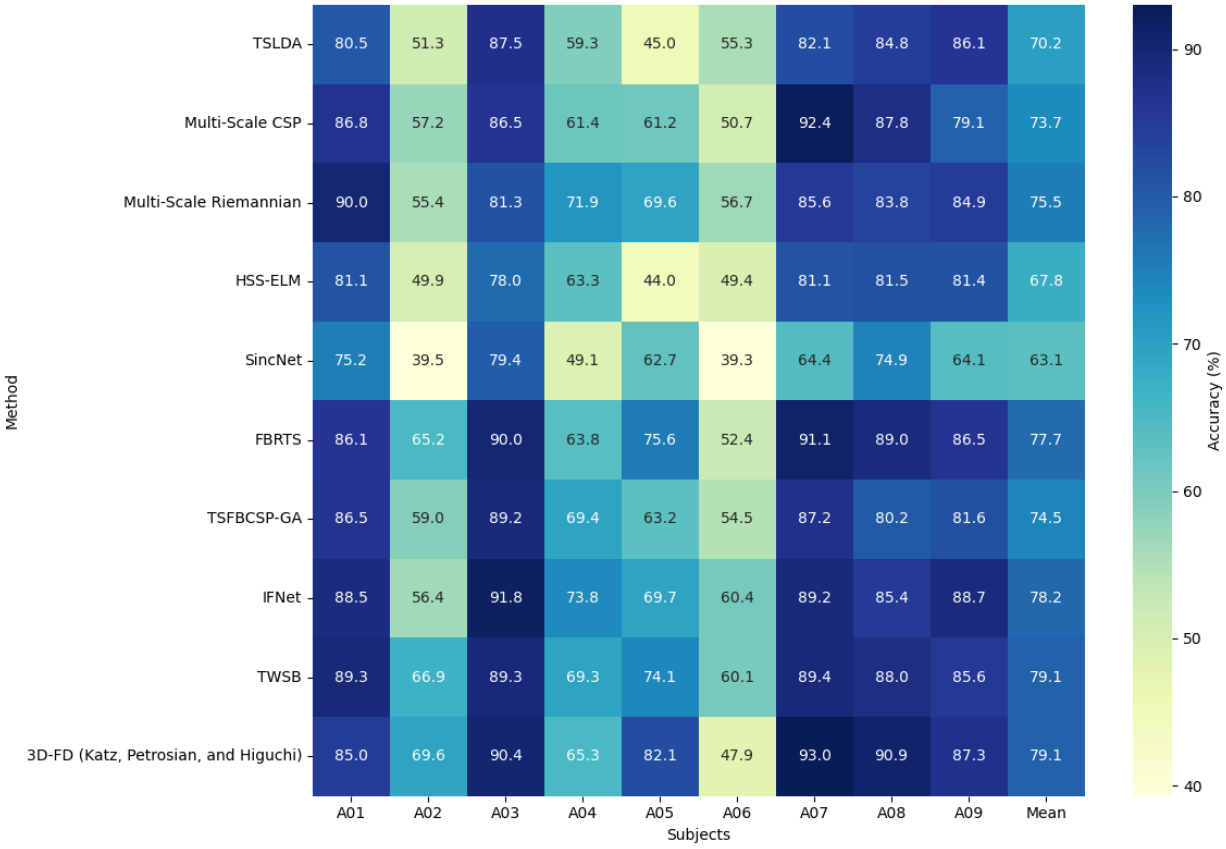

Figure 5. A heatmap of the classification accuracy of various methods across subjects and the overall mean.

After analyzing the subject accuracy for our method, 3D-FD, compared to the other approaches, the results show that 3D-FD achieves high and consistent classification accuracy across subjects, with six out of nine subjects showing an accuracy above 80%. The balanced performance, without extreme highs or lows, indicates that 3D-FD generalizes well across diverse subjects despite inter-subject EEG variability concerns. This stability is verified by the overall mean accuracy of 79.1%, which aligns with the best MI method year to date, the TWSB method [45].

Student's t-tests were performed again to compare the 3D-FD method with various existing methods. A significant difference was found between 3D-FD and both HSS-ELM ($p = 0.0212$) and SincNet ($p = 0.0001$), with 3D-FD performing significantly better in both cases. No significant differences were found between 3D-FD and the other methods (all p-values > 0.05). These results indicate that while 3D-FD shows a noted advantage over certain methods, like HSS-ELM and SincNet, its performance is comparable to other existing methods (see Table 11).

The significant differences observed between the 3D-FD method and both HSS-ELM and SincNet highlight that 3D-FD can offer advantages over these approaches, possibly due to its unique feature extraction capabilities in capturing complex data patterns from EEG signals. The lack of significant differences from other methods shows that 3D-FD performs comparably to existing approaches. The latter highlights the robustness of 3D-FD without significant statistical superiority.

Table 11. Student's *t*-test findings for the 3D-FD versus other methods on dataset 2a.

Comparison	t-Statistic	*p*-Value
3D-FD vs. TSLDA	2.0942	0.0696
3D-FD vs. Multi-Scale CSP	2.1535	0.0634
3D-FD vs. Multi-Scale Riemannian	1.2626	0.2423
3D-FD vs. HSS-ELM	2.8574	0.0212
3D-FD vs. SincNet	6.8718	0.0001
3D-FD vs. FBRTS	1.2607	0.2429
3D-FD vs. TSFBCSP-GA	1.6665	0.1342
3D-FD vs. IFNet	0.2899	0.7793
3D-FD vs. TWSB	−0.0282	0.9782

6. Discussion

In this section, we discuss the associated results and conduct a thorough analysis of the outcomes obtained from the study, offering insights into their significance and exploring any possible future study areas.

The results obtained in this study benefit the field of motor imagery. The outcomes show the efficacy of the LinearSVM classification method, which consistently delivered the best overall performance and had the highest success rate among most feature sets in terms of fractal dimensions and statistical features. This high performance can be linked to the classifier's ability to handle highly dimensional data while minimizing the risk of overfitting. GSVM is in second place and sometimes performs better on specific feature sets, like fractal dimensions and wavelet analysis, and sometimes struggles with others, like statistical and all combined. The third place is attributed to SVM with a polynomial kernel. It also performs better with fractal dimensions and wavelet analysis and lags behind other feature sets. GSVM and SVM with a polynomial kernel show moderate performance. As they are non-linear SVMs, they offer an easy way of capturing complex relationships in EEG signals. However, they may be more disposed to overfitting with highly dimensional features, and the latter can affect the consistency of the results when using different feature extraction techniques. Lastly, CART remains the least effective classifier, with a low average success rate across all five feature sets except statistical features. This is likely because CART is sensitive to noise and tends to overfit with complex, highly dimensional data. While CART is usually used for non-linear decision making, this sensitivity can lead to a lower accuracy when applied to multi-class MI tasks, where the data can vary.

Considering the feature extraction methods used, three-dimensional fractal dimensions (Katz, Petrosian, and Higuchi) outperform all other feature sets, except in the case of use of the CART classifier, where the statistical method outperforms fractal dimensions. Moreover, all combined features led to lower classification accuracy, suggesting that by combining different feature sets, extra noise is generated, which makes it harder for the classifiers to identify the differences in the EEG signals. This can highlight a potential limitation of combining features.

The sequential preprocessing approach adopted in this study is effective for noise reduction and feature extraction in motor imagery EEG signals, but there are alternative techniques, such as regression-based filtering [46] and independent component analysis (ICA) [35], which can enhance filtering artifact corrections, specifically in the case of motion noise or muscle movements. Despite the latter, these methods may create more complexity and computational demands, which is beyond the scope of this current research.

During our previous study [47], we only utilized the binary classification of left and right classes using the BCI Competition IV 2a dataset and all five feature sets. Meanwhile, a combination of only three feature sets (fractal dimensions, wavelet analysis, and Hjorth) was implemented. However, in this latest study, all four motor imagery classes were used, left and right hands, both feet, and tongue, and all combined (statistical, wavelet analysis, higher-order spectra, Hjorth, and fractal dimension) features were employed. Unlike GSVM and CART, LinearSVM and SVM with a polynomial kernel were added.

When relating to the results obtained in the motor imagery reference paper [14], which featured our MI pipeline, traditional feature extraction techniques such as common spatial patterns (CSPs) and Riemannian Geometry-based approaches have been widely used for motor imagery classification. The latter still face some limitations, like handling low signal-to-noise ratios and capturing complex, non-linear dynamics characteristics of EEG signals. For example, CSP is highly sensitive to noise and primarily optimized for binary classification, making it less suitable for multi-class scenarios similar to our use case. Yet, the ER-based features employed, fractal dimensions (Katz, Petrosian, Higuchi), wavelet analysis, and higher-order spectra (HOS), are designed to handle these types of issues, underlining the superior robustness, as shown by the remarkable performance gained in this study.

Our mixed ER/MI approach using three-dimensional fractal dimensions (Katz, Petrosian, and Higuchi) reached an exact accuracy of 79.04% for LinearSVM, which is even higher than the top-of-the-line results acquired, regardless of CSP (73.70%) or Riemannian (75.47%) [14]. Moreover, the current findings align with the emotion recognition findings adopted from [7], as the latter stated that the best overall accuracy was achieved using the three-dimensional fractal dimension feature set. In this study, fractal dimensions showed promising results when applied to the motor imagery span. Furthermore, when compared with the latest method used in the MI field, the TWSB method from [45], our results reached approximately the same accuracy of 79.1%.

The current research presents novel findings, specifically the greater accuracy of fractal dimension features, including Katz, Petrosian, and Higuchi. It also highlights the improved performance of LinearSVM compared to GSVM, CART, and SVM with a polynomial kernel. As fractal dimension features are robust to noise and capture notable structural patterns within the EEG signals used, this leads to an enhancement in the system's ability to classify MI tasks with subtle signal differences. This makes this custom pipeline distinctive and underlines a unique contribution to the classification improvements of the motor imagery field with feature sets from the emotion recognition field.

In future research, more fractal dimension algorithms can be introduced to our embraced approach. By combining methods other than the Katz, Petrosian, and Higuchi algorithms, different groupings might yield different outcomes. By extending our mixed pipeline, comparing the latter might influence the overall accuracy precisely when different fractal dimensions are used with the four performance classifiers utilized in the current research: GSVM, CART, LinearSVM, and SVM with a polynomial kernel. This extension aims for even higher accuracy and better-quality average success rates.

7. Conclusions

Our study proposes a new approach that contributes to enhancing the MI-BCI-related domain of research. This approach combines methods from both emotion recognition and motor imagery. As this study uses all four classes of motor imagery tasks, the analysis of the EEG data is multi-classified instead of the use of binary classification. The addition of a five-dimensional combination of feature sets (statistical, wavelet, HOS, Hjorth, and fractal dimensions) showed that when merging the methods, noise was generated, leading to less accuracy. In addition to our previous study's classifiers, GSVM and CART, two more were introduced: LinearSVM and SVM with a polynomial kernel. This custom hybrid pipeline delivers noteworthy results. The fractal dimensions method, using the Katz, Petrosian, and Higuchi algorithms, achieves superior accuracy in motor imagery tasks, reaching $\approx$ 79.1%. Regarding classifiers implemented on the BCI Competition IV-2a dataset, LinearSVM regularly accentuates better accuracy than GSVM, CART, and SVM with a polynomial kernel. Our novel method results were confirmed by the referenced emotion recognition and motor imagery articles, as 3D-FD combined with LinearSVM highlighted an increase in the success rate and overall average accuracy. Overall, the findings generated from this study can help create more robust and effective BCI systems, expanding the possibilities for real-world applications. This emphasizes that applying feature extraction

and classifier methods in the emotion recognition field to the motor imagery field yields significant results.

Author Contributions: V.J. supervised the research, analyzed the results, provided feedback, revised the draft, and approved the final version of the article. A.F.M. implemented the research, executed experimental work, analyzed the results, and revised the final article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the BCI Competition IV 2a, https://www.bbci.de/competition/iv/, accessed on 13 March 2024 [15].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BCI	Brain–computer interface.
CAR	Common average reference.
CART	Classification and regression tree.
CLI	Command Line Interface.
CSP	Common spatial pattern.
DWT	Discrete wavelet transform.
EEG	Electroencephalogram.
EOG	Electrooculography.
ER	Emotion recognition.
ER-EEG	Emotion Recognition Electroencephalogram.
FD	Fractal dimensions.
fMRI	Functional magnetic resonance imaging.
fNIRS	Functional near-infrared spectroscopy.
GDF	General Data Format.
GSVM	Gaussian support vector machine.
GUI	Graphical user interface.
HOS	Higher-order spectra.
HPFs	High-Pass Butterworth Filters.
ICA IDE	Independent component analysis. Integrated development environment.
MAT file	MATLAB file.
MI	Motor imagery.
MI-EEG	Motor Imagery Electroencephalogram.
Poly	Polynomial.
PSD	Power spectral density.
RBF	Radial basis function.
SNR	Signal-to-noise ratio.
SVM	Support vector machine.

References

1. DeCharms, R.C.; Christoff, K.; Glover, G.H.; Pauly, J.M.; Whitfield, S.; Gabrieli, J.D. Learned regulation of spatially localized brain activation using real-time fMRI. *Neuroimage* **2004**, *21*, 436–443. [CrossRef] [PubMed]
2. Coyle, S.M.; Ward, T.E.; Markham, C.M. Brain–computer interface using a simplified functional near-infrared spectroscopy system. *J. Neural Eng.* **2007**, *4*, 219. [CrossRef] [PubMed]
3. Baillet, S.; Mosher, J.C.; Leahy, R.M. Electromagnetic brain mapping. *IEEE Signal Process. Mag.* **2001**, *18*, 14–30. [CrossRef]
4. Singh, A.; Lal, S.; Guesgen, H.W. Architectural review of co-adaptive brain computer interface. In Proceedings of the 2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Melbourne, Australia, 11–13 December 2017; IEEE: New York, NY, USA; pp. 200–207.

5. Wolpaw, J.R.; Birbaumer, N.; McFarland, D.J.; Pfurtscheller, G.; Vaughan, T.M. Brain–computer interfaces for communication and control. *Clin. Neurophysiol.* **2002**, *113*, 767–791. [CrossRef]
6. Houssein, E.H.; Hammad, A.; Ali, A.A. Human emotion recognition from EEG-based brain–computer interface using machine learning: A comprehensive review. *Neural Comput. Appl.* **2022**, *34*, 12527–12557. [CrossRef]
7. Yuvaraj, R.; Thagavel, P.; Thomas, J.; Fogarty, J.; Ali, F. Comprehensive analysis of feature extraction methods for emotion recognition from multichannel EEG recordings. *Sensors* **2023**, *23*, 915. [CrossRef]
8. McFarland, D.J.; Wolpaw, J.R. EEG-based brain–computer interfaces. *Curr. Opin. Biomed. Eng.* **2017**, *4*, 194–200. [CrossRef]
9. Wei, Q.; Zhu, S.; Wang, Y.; Gao, X.; Guo, H.; Wu, X. Maximum signal fraction analysis for enhancing signal-to-noise ratio of EEG signals in SSVEP-based BCIs. *IEEE Access* **2019**, *7*, 85452–85461. [CrossRef]
10. Qin, X.; Zheng, Y.; Chen, B. Extract EEG features by combining power spectral density and correntropy spectral density. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; IEEE: New York, NY, USA; pp. 2455–2459.
11. Katz, M.J. Fractals and the analysis of waveforms. *Comput. Biol. Med.* **1988**, *18*, 145–156. [CrossRef]
12. Hatamikia, S.; Nasrabadi, A.M. Recognition of emotional states induced by music videos based on nonlinear feature extraction and SOM classification. In Proceedings of the 2014 21st Iranian Conference on Biomedical Engineering (ICBME), Tehran, Iran, 26–28 November 2014; IEEE: New York, NY, USA; pp. 333–337.
13. Higuchi, T. Approach to an irregular time series on the basis of the fractal theory. *Phys. D Nonlinear Phenom.* **1988**, *31*, 277–283. [CrossRef]
14. Hersche, M.; Rellstab, T.; Schiavone, P.D.; Cavigelli, L.; Benini, L.; Rahimi, A. Fast and accurate multiclass inference for MI-BCIs using large multiscale temporal and spectral features. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; IEEE: New York, NY, USA; pp. 1690–1694.
15. Brunner, C.; Leeb, R.; Müller-Putz, G.; Schlögl, A.; Pfurtscheller, G. BCI Competition 2008–Graz data set A. *Inst. Knowl. Discov. (Lab. Brain-Comput. Interfaces), Graz Univ. Technol.* **2008**, *16*, 1–6.
16. Singh, A.; Hussain, A.A.; Lal, S.; Guesgen, H.W. A comprehensive review on critical issues and possible solutions of motor imagery based electroencephalography brain-computer interface. *Sensors* **2021**, *21*, 2173. [CrossRef] [PubMed]
17. Solé-Casals, J.; Caiafa, C.; Zhao, Q.; Cichocki, A. Brain-Computer Interface with Corrupted EEG Data: A Tensor Completion Approach. *Cognit. Comput.* **2018**, *10*, 1062–1074. [CrossRef]
18. Gaur, P.; Pachori, R.B.; Wang, H.; Prasad, G. An Automatic Subject Specific Intrinsic Mode Function Selection for Enhancing Two-Class EEG-Based Motor Imagery-Brain Computer Interface. *IEEE Sens. J.* **2019**, *19*, 6938–6947. [CrossRef]
19. Togha, M.M.; Salehi, M.R.; Abiri, E. Improving the performance of the motor imagery-based brain-computer interfaces using local activities estimation. *Biomed. Signal Process. Control* **2019**, *50*, 52–61. [CrossRef]
20. Sampanna, R.; Mitaim, S. Noise benefits in the array of brain-computer interface classification systems. *Inform. Med. Unlocked* **2018**, *12*, 88–97. [CrossRef]
21. Feng, J.K.; Jin, J.; Daly, I.; Zhou, J.; Niu, Y.; Wang, X.; Cichocki, A. An Optimized Channel Selection Method Based on Multifrequency CSP-Rank for Motor Imagery-Based BCI System. *Comput. Intell. Neurosci.* **2019**, *2019*, 8068357. [CrossRef]
22. Ramakrishnan, A.; Satyanarayana, J. Reconstruction of EEG from limited channel acquisition using estimated signal correlation. *Biomed. Signal Process. Control* **2016**, *27*, 164–173. [CrossRef]
23. Hirano, K.; Nishimura, S.; Mitra, S. Design of digital notch filters. *IEEE Trans. Commun.* **1974**, *22*, 964–970. [CrossRef]
24. Hussin, S.F.; Birasamy, G.; Hamid, Z. Design of Butterworth band-pass filter. *Politeknik Kolej Komuniti J. Eng. Technol.* **2016**, *1*, 32–46.
25. Lemos, M.S.; Fisch, B.J. The weighted average reference montage. *Electroencephalogr. Clin. Neurophysiol.* **1991**, *79*, 361–370. [CrossRef] [PubMed]
26. Jenke, R.; Peer, A.; Buss, M. Feature extraction and selection for emotion recognition from EEG. *IEEE Trans. Affect. Comput.* **2014**, *5*, 327–339. [CrossRef]
27. Wang, X.W.; Nie, D.; Lu, B.L. Emotional state classification from EEG data using machine learning approach. *Neurocomputing* **2014**, *129*, 94–106. [CrossRef]
28. Hosseini, S.A. Classification of brain activity in emotional states using HOS analysis. *Int. J. Image Graph. Signal Process.* **2012**, *4*, 21. [CrossRef]
29. Hjorth, B. EEG analysis based on time domain properties. *Electroencephalogr. Clin. Neurophysiol.* **1970**, *29*, 306–310. [CrossRef]
30. Hjorth, B. The physical significance of time domain descriptors in EEG analysis. *Electroencephalogr. Clin. Neurophysiol.* **1973**, *34*, 321–325. [CrossRef]
31. Yang, J.; Wu, Z.; Peng, K.; Okolo, P.N.; Zhang, W.; Zhao, H.; Sun, J. Parameter selection of Gaussian kernel SVM based on local density of training set. *Inverse Probl. Sci. Eng.* **2021**, *29*, 536–548. [CrossRef]
32. Azuaje, F.; Witten, I.H.; Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed.; Morgan Kaufmann Publishers: San Francisco, CA, USA, 2005; p. 560. ISBN 0-12-088407-0.
33. Hsu, C.W. *A Practical Guide to Support Vector Classification*; Department of Computer Science, National Taiwan University: Taipei, Taiwan, 2003.
34. Diamantidis, N.A.; Karlis, D.; Giakoumakis, E.A. Unsupervised stratification of cross-validation for accuracy estimation. *Artif. Intell.* **2000**, *116*, 1–6. [CrossRef]

35. Kim, H.; Luo, J.; Chu, S.; Cannard, C.; Hoffmann, S.; Miyakoshi, M. ICA's Bug: How Ghost ICs Emerge from Effective Rank Deficiency Caused by EEG Electrode Interpolation and Incorrect Re-Referencing. *Front. Signal Process.* **2023**, *3*, 1064138. [CrossRef]
36. Lotte, F.; Guan, C. Regularizing Common Spatial Patterns to Improve BCI Designs: Unified Theory and New Algorithms. *IEEE Trans. Biomed. Eng.* **2010**, *58*, 355–362. [CrossRef]
37. Miladinović, A.; Ajčević, M.; Jarmolowska, J.; Marusic, U.; Silveri, G.; Battaglini, P.P.; Accardo, A. Performance of EEG Motor-Imagery Based Spatial Filtering Methods: A BCI Study on Stroke Patients. *Procedia Comput. Sci.* **2020**, *176*, 2840–2848. [CrossRef]
38. Zhang, Y.; Zhou, G.; Jin, J.; Wang, X.; Cichocki, A. Optimizing Spatial Patterns with Sparse Filter Bands for Motor-Imagery Based Brain–Computer Interface. *J. Neurosci. Methods* **2015**, *255*, 85–91. [CrossRef] [PubMed]
39. Izzuddin, T.A.; Safri, N.M.; Othman, M.A. Compact convolutional neural network (CNN) based on SincNet for end-to-end motor imagery decoding and analysis. *Biocybern. Biomed. Eng.* **2021**, *41*, 1629–1645. [CrossRef]
40. She, Q.; Hu, B.; Luo, Z.; Nguyen, T.; Zhang, Y. A hierarchical semi-supervised extreme learning machine method for EEG recognition. *Med. Biol. Eng. Comput.* **2019**, *57*, 147–157. [CrossRef]
41. Wang, J.; Yao, L.; Wang, Y. IFNet: An interactive frequency convolutional neural network for enhancing motor imagery decoding from EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2023**, *31*, 1900–1911. [CrossRef]
42. Barachant, A.; Bonnet, S.; Congedo, M.; Jutten, C. Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 920–928. [CrossRef]
43. Luo, T.-J. Parallel genetic algorithm based common spatial patterns selection on time-frequency decomposed EEG signals for motor imagery brain-computer interface. *Biomed. Signal Process. Control* **2023**, *80*, 104397.
44. Fang, H.; Jin, J.; Daly, I.; Wang, X.Y. Feature extraction method based on filter banks and Riemannian tangent space in motor-imagery BCI. *IEEE J. Biomed. Health Inf.* **2022**, *26*, 2504–2514. [CrossRef]
45. Li, Z.; Tan, X.; Li, X.; Yin, L. Multiclass motor imagery classification with Riemannian geometry and temporal-spectral selection. *Med. Biol. Eng. Comput.* **2024**, *62*, 1–3. [CrossRef]
46. Reddy, T.K.; Arora, V.; Behera, L.; Wang, Y.K.; Lin, C.T. Multiclass Fuzzy Time-Delay Common Spatio-Spectral Patterns with Fuzzy Information Theoretic Optimization for EEG-Based Regression Problems in Brain–Computer Interface (BCI). *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1943–1951. [CrossRef]
47. Mohamed, A.F.; Jusas, V. Comprehensive analysis of feature extraction methods for emotion recognition on motor imagery from multichannel EEG recordings. In Proceedings of the World Conference on Information Systems and Technologies, Funchal, Portugal, 26–28 March 2024; Springer Nature: Cham, Switzerland; pp. 211–231.