scientific reports

Check for updates

OPEN

Error-related potentials in EEG signals: feature-based detection for human-robot interaction

Alessandra Fava✉, Valeria Villani & Lorenzo Sabattini

This study explores how to improve the detection of Error-Related Potentials (ErrPs), namely brain signals generated when a person perceives an unexpected action performed by an interacting agent. ErrPs are promising for improving interactions between humans and robots because they offer a way for robots to understand the user's needs and expectations without explicit input. The proposed method aims at characterizing ErrP signals using a wide set of features extracted from electroencephalography (EEG) data, collected from subjects performing different tasks. This feature-based method results more accurate and efficient than traditional approaches, especially when applied to multiple users, or across different experimental setups. This work paves the way to feature-based ErrP detection to enhance human-robot interaction in dynamic environments.

Keywords EEG, ErrP, Machine learning, Selection of features, Cross-subjects classification

The widespread adoption of robots is transforming industries and everyday life, enhancing efficiency and precision in manufacturing, healthcare, and even household tasks[1–6]. To facilitate and smoothen interaction, it is crucial that robotic agents can understand users' needs and preferences and adapt accordingly. This applies to different contexts, ranging from industrial shopfloors where a single robot is used by different workers across different shifts[1–3], to assistive robots in domestic environments involving various individuals such as elderly users, family members, and caregivers[4–6]. There are various methods for a user to communicate their needs and preferences to a robot. For instance, voice control allows for hands-free interaction but may be unsuitable in noisy environments. Joysticks or keyboards are commonly used but require physical effort and proximity. Moreover, these modalities require users to explicitly express their intentions, which can be impractical as it interrupts the interaction flow and may be prone to misinterpretation.

To overcome these limitations, the use of implicit communication (i.e., involuntary or not explicitly verbalized by the user) and wearable devices could be a way for the user to provide a signal to the robot. Implicit communication can be provided, for instance, by means of physiological signals, which include heart rate, blood pressure, electrocardiogram, and electroencephalogram (EEG). While they are generally used to monitor human health, they can also be leveraged for the purpose of human-robot interaction (HRI)[7–11]. In particular, we focus on the measurement of a specific kind of brain activity, called Error-Related Potential (ErrP), recorded through EEG. ErrPs are a specific type of Event-Related Potentials (ERPs), which are evoked in response to external stimuli or specific perceptual, cognitive, or motor events[12]. Specifically, ErrPs are evoked when a stimulus is perceived as incorrect. As a consequence, we leverage them to provide an implicit feedback to the robot. In particular, when a subject perceives an incorrect decoding of their intention by the robot, an ErrP signal will be elicited in their brain (Fig. 1). This also occurs when a robot's movement does not correspond to the command sent by the subject to the robot. By detecting ErrPs, it is possible to provide a feedback from the human to the robot, which alerts the robot about an incorrect movement. In this way, the robot can take corrective actions and learn from its errors[13,14].

The ErrP signal is characterized by a specific waveform composed of different peaks with a defined latency. A positive (P) peak, called P50, can be observed around 50 msec after the stimulus, followed by a negative peak after 100 msec (N100), a P200, N300, P300, N400, and P600[15]. Previous studies suggest that ErrPs are generated by brain regions involved in error processing, with the anterior cingulate cortex (ACC) being a likely neural source, as indicated by converging electrophysiological and neuroimaging evidence[16]. An alternative characterization of ErrP is defined by a negative potential deflection, called Error-Related Negativity (ERN), that appears over fronto-central scalp areas at about 50–100 ms after an unexpected event[17,18]. This negative component is followed by a centro-parietal positive deflection (Pe) at about 200 to 300 ms[19].

Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, 42122 Reggio Emilia, Italy. ✉email: alessandra.fava@unimore.it

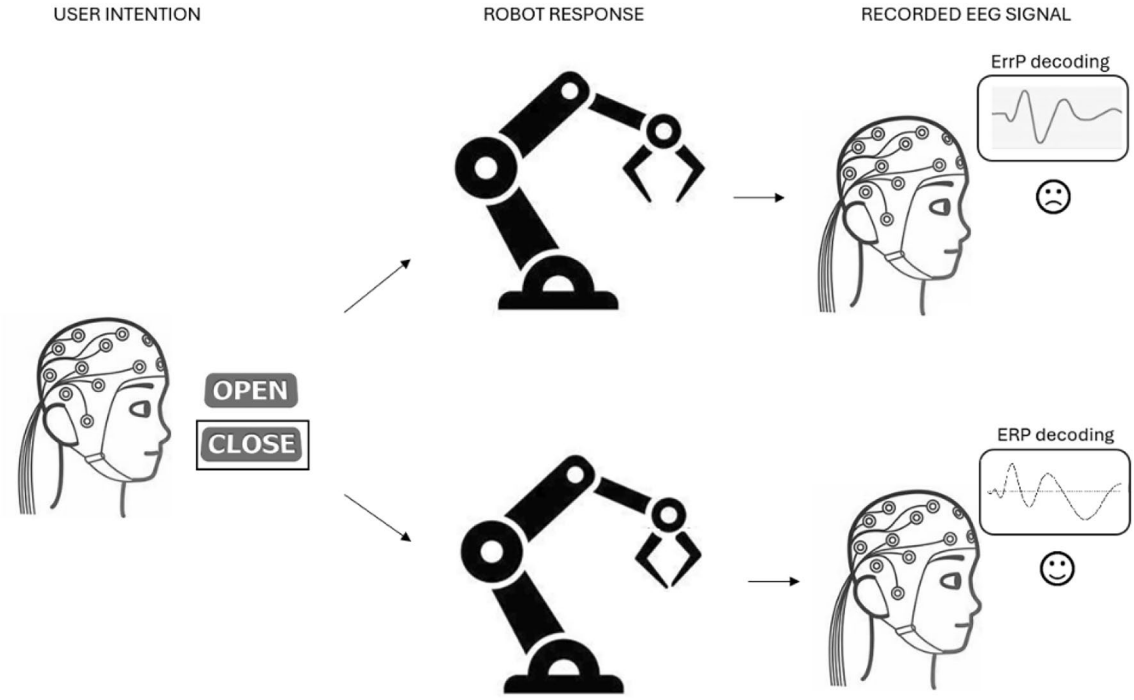

Fig. 1. Example of ErrP detection for brain-robot interaction: the user selects "close" for the robotic gripper, but if the gripper remains open, an ErrP is detected in the EEG signal. Instead, if the gripper closes (which is the movement wanted by the user), then no ErrP arises, but a simple ERP is evoked.

Generally, ErrPs are observed in four distinct task scenarios[13,14]:

- *Interaction*: ErrPs emerge when errors occur during human-machine interaction;
- *Feedback*: ErrPs arise when errors are perceived in the command given by the subject;
- *Response*: ErrPs occur when errors are made by the subject in rapid response tasks, such as choice reaction tasks;
- *Observation*: ErrPs are triggered when errors are observed in actions executed by a system or robot, during an HRI task.

In this work, we focus on the use of ErrPs in feedback task scenarios.

Related work and contribution

The use of ErrPs in HRI is reported in the literature in different contexts. For instance, they were used to control a wheelchair[20], a prosthetic hand[21], an exoskeleton system[22,23] or to play virtual reality games[24]. In their study[25], Jeong et al. introduced a prototype of a brain-controlled robot arm system, employing various upper limb Motor Imagery (MI) paradigms. Lin et al.[26] devised a wireless steady-state visually evoked potential (SSVEP)-based brain-robot interface (BRI) system for controlling a robotic arm, successfully aiding patients in self-feeding. Meanwhile, Nurseitov et al.[27] utilized P300 potentials to steer a mobile robot in four directions. Both SSVEP and MI-based BRIs have shown high performance in various standard robotic control tasks[28,29]. Salazar-Gomez et al.[30] proposed a closed-loop system leveraging ErrPs as implicit inputs to guide a robotic arm in binary bin-sorting tasks. Kim et al.[14] employed ErrPs as implicit rewards for a real robotic system to learn associations between human gestures and corresponding actions. Furthermore, *Ehrlich and Cheng*[31] showcased the utility of ErrPs as feedback signals from humans for facilitating real-time adaptive interaction in HRI. Lopes-Dias et al.[32] showed that online asynchronous decoding of ErrPs can be used as feedback to guide a robotic arm toward a target, even under unexpected interruptions.

However, although many studies demonstrate the usefulness of ErrPs in BRI, detecting these signals remains challenging. The neural helmets typically used for EEG recording are uncomfortable and require time consuming setup. The experiments are often long in duration because it is necessary to record large amounts of data to enhance signal quality. Moreover, the recording process may be very sensitive to artifacts and noise and, hence, requires extreme care[33]. Artifacts are external signals, not related to cerebral activity: they could be electrical, physiological, or environmental, usually due to eye movement and blinks, myogenic activity, cardiac activity, chewing and hypoglossal movements[34]. Hence, one of the first essential steps to improve the recording process is to remove noise and artifacts during EEG processing with advanced filtering and feature extraction techniques. In addition, the quality of recorded signals can be increased using higher performance sensors.

In order to use ErrPs as implicit feedback in HRI, the aim of this work is to improve the effectiveness of ErrP detection. Specifically, we consider the case of a robot that performs different tasks interacting with and adapting

to different users. Hence, we propose a general approach for the detection of ErrPs in EEG signals recorded from different subjects and/or performing different HRI tasks.

Our focus is on cross-subject classification, with the goal of developing a subject-independent method that enables reliable ErrP detection across users. This capability is particularly relevant in real-world applications where a single robot may be used by multiple individuals–for instance, in industrial settings with rotating shifts, or domestic environments shared by several people. The novelty of our work lies in the development of a unified and feature-based classification strategy that generalizes across subjects and tasks without the need for subject-specific calibration. This approach addresses a major bottleneck in the deployment of ErrP-based BCIs in real-world applications, supporting scalable and adaptive HRI. ErrP detection involves distinguishing between brain response when there is a correct event (ERP), and brain response when a robot error is perceived by the user (ErrP), using classification models. In order to evaluate the performance of the classification models, the most commonly used metrics are accuracy and recall[7,10,35–40]. Accuracy measures the ratio of correctly predicted instances (both ErrP and non-ErrP) over the total number of instances, calculated as

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})},$$

where:

- TP (True Positives): correctly predicted ErrP trials.
- TN (true negatives): correctly predicted non-ErrP (ERP) trails.
- FP (false Positives): non-ErrP (ERP) trails incorrectly classified as ErrP.
- FN (False Negatives): ErrP trails incorrectly classified as non-ErrP (ERP).

However, since the datasets used in this study are imbalanced, with significantly more non-ErrP samples compared to ErrP samples, classification performance cannot be reliably assessed using accuracy alone. In this context, we report additional metrics such as recall (also known as true positive rate or sensitivity), which quantifies the proportion of correctly detected ErrP trials:

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}.$$

This metric is particularly relevant for implicit communication in HRI, such as P300 classification[41], where failing to detect a perceived error (i.e., a false negative) could prevent corrective feedback, potentially compromising task execution. On the contrary, a false positive may simply lead to an unnecessary correction, which is generally less disruptive[21,24].

To ensure a more balanced performance evaluation under class imbalance, we also adopt the F1 score[42], a metric that captures the trade-off between precision and recall and is widely recognized in imbalanced classification contexts. The F1 score is the harmonic mean of precision and recall:

$$\text{F1} = \frac{2(\text{Precision} \cdot \text{Recall})}{(\text{Precision} + \text{Recall})}$$

where

$$\text{Precision} = \frac{\text{TP}}{(\ TP + FP)}.$$

The F1 score provides a more comprehensive view of model performance, particularly when both false positives and false negatives are consequential, as is the case in real-time ErrP detection for HRI.

To address class imbalance, we applied simple oversampling by duplicating minority class instances (i.e., ErrP trials). This method was chosen over undersampling–which risks discarding valuable information from the majority class–and over synthetic techniques like SMOTE, which are less commonly used for EEG data due to the difficulty in generating realistic synthetic signals that retain essential temporal, spatial, and spectral properties[11,43].

While oversampling may inflate recall, we justify this choice by the critical importance of error detection in real-time applications. To mitigate potential bias, we rely on the F1 score as a balanced metric that accounts for both precision and sensitivity. Future work will explore more advanced, EEG-specific data augmentation techniques, including the use of generative models.

Finally, to enable real-time detection in dynamic environments, the system must adapt to new users and update its capabilities over time. This is especially important in scenarios where robot-user interactions evolve, requiring the detection model to incorporate incremental learning strategies. Such strategies allow the model to be initially trained on a limited dataset and then continuously updated as new data become available–without the need for complete retraining, which can be computationally costly for advanced detection models.

Building upon these lines, the contribution of this work is an approach that satisfies the following requirements:

R1: *high accuracy, recall and F1*, to validate the effectiveness and the quality of detection;

R2: *low time to build the model*, to enable real-time incremental learning as new data are available.

While customarily ErrP detection is performed considering all the samples of EEG recordings, in this work we propose to detect ErrPs using specific characteristics of the EEG signal, referred to as *features*. These features

were analyzed in our previous work[44], and allow the extraction of information from an ErrP signal, enhancing signal explainability. They include a set of 71 features, composed of 12 temporal features, 8 frequency features, 4 processing features, 2 statistical features, 17 coefficients of Wavelet transform in the [2-4] Hz band and 28 coefficients of Wavelet transform in [4-8] Hz band. More details are given in the *Methods* section. Therefore, our approach starts from the extraction of the features and evaluates their contribution to ErrP detection, in terms of the two requirements described above. As opposed to the use of all the samples of EEG signals, the use of these features characterizes the ErrP signal, while retaining signal explainability, embedding variability among individuals, and reducing the time required to build a model. Our idea is that the features allow to represent the intrinsic characteristics of ErrPs, discarding variability due to subject baseline brain activity and ERP signals, thus providing models that are less dependent on the subject from which they were trained. This approach paves the way for incremental learning scenarios, where data from new subjects can be classified using models previously built on other subjects and consequently updated.

Results

In our previous work[44], features were introduced for classification, and a statistical analysis was reported on their relevance on ErrP detection. As will be described in details in the *Methods* section, features are divided in groups based on their relevance. Analyzing the accuracy obtained with the different groups of features, we selected the following four groups: 14 features, 21 features, 28 features, all the features (71). The results using the groups of features were compared with those achieved using all the samples within an epoch. In this section, we analyze the use of the features exploiting several machine learning models commonly used in the literature. We consider cross-subject classification across different datasets to show the generality of the proposed ErrP detection approach. In addition, in the *Supplementary results* paragraph, we provide additional results related to single-subject classification and cross-subject classification within dataset.

The approach was tested considering two datasets. The first one is introduced and analyzed in[45]. Data were collected with experiments involving 11 subjects, performed in two different scenarios, called "cursor scenario" and "robot scenario". Instead, the second dataset is described in[18]. It includes data collected with experiments involving six subjects, performed in two sessions, labeled as "session 1" and "session 2". The datasets are described in detail in the *Methods* paragraph. The datasets are split into training and testing sets, allocating 80% of the data for training and 20% for testing.

In this work, we do not address the pre-processing phase: in fact, we exploit the commonly used processing pipeline described in[45]. It is worth noting that we need to deal with an unbalanced dataset, because the number of correct trials is larger than the number of ErrP signals. For this purpose, we introduce an oversampling technique[46], duplicating the instances of the trials that present ErrPs. To assess performance, we evaluate the following metrics[47]: accuracy, recall and and F1, where high values are desirable, and time to build the model, where low values are desirable. Specifically, for the sake of comparison, time to build the model is expressed in terms of a *time complexity score*, which takes the values in Table 1.

As regards the considered classification methods, according to the literature, classification is performed using the following machine learning models: decision tree (DT)[48], k-nearest neighbors (KNN)[49], support vector machine (SVM)[49], linear discriminant analysis (LDA)[49], naive Bayes (NB)[50], and ensemble learning method (ELM)[51]. For each model and feature group, both oversampling and non-oversampling configurations were evaluated to assess the impact of data balancing on classification performance for all the aforementioned models. To maintain clarity and conciseness in the presentation of results, only the configuration yielding the best performance is reported in the tables. This approach avoids overloading the results section with excessive experimental variations while highlighting the most effective setups, to compare them with the performance of the method usually used in the literature, that involves the use of all samples (358 samples for each channel, for both dataset).

As mentioned above, we focus on cross-subject classification, meaning that training and testing sets are built including data from different subjects. Furthermore, since we consider two datasets, with a total of four different tasks ("cursor scenario" and "robot scenario" and "session 1" and "session 2"), training and testing sets are built both merging and combining tasks. In the first case, we perform cross-subjects classification within datasets: both training and testing data are selected from both datasets, considering different subjects performing both the tasks. In the second case we combine data taken from the two datasets in a between design: data from the training set are taken from a dataset and data for the testing set are taken from the other. The results of these

Time complexity score	Time to build the model
1	Less than 1 minute
2	From 1 minute to 10 minutes
3	From 10 minutes to 1 hour
4	From 1 hour to 3 hours
5	From 3 hours to 6 hours
6	From 6 hours to 12 hours
7	From 12 hours to 1 day
8	More than 1 day

Table 1. Time complexity score index, to identify the different time to build the model.

analyses are reported in Tables 2, 3 and 4, where, in each of them, we summarize the configuration of training and testing sets (light blue tables).

Cross-subject classification within dataset

In this paragraph, we merge the data acquired in the two sessions of both datasets. Our aim is to assess the generalization of the proposed tasks under different configurations, while keeping task and scenario conditions consistent between training and testing sets. In particular, we train the model on 26 subjects, randomly selected extracting nine of them from each session of[45], and four of them from each session of[18]. The model is then tested on the remaining eight subjects (two from each session of each dataset).

Results are reported in Table 2, and, at the top of the table, we report the organization of the training and testing sets, used to train the models and assess their performance. The **R1** requirement is partially met by two different models. The best accuracy and F1 are obtained with the all signal samples model (89.4% and 92.6% respectively) and the best recall is obtained with the all features model (99.1%). While the model using all features achieves the highest classification performance, time constraints play a crucial role in our application. As such, models based on 14 and 21 features offer a better balance between performance and processing time. Furthermore, the model built with 21 features presents the best time complexity score, thus meeting the **R2** requirement.

In this case, we obtained that the **R2** requirement is always met using the feature-based models, and the **R1** requirement is partially met using the all features model (in recall) and with the all signal samples model (in accuracy and F1).

Cross-subject classification between datasets

In this paragraph we train the models on data from dataset[45] and test them on data from dataset[18], considering all the possible combinations of scenarios. In details, training is performed considering three possible scenarios: *(i)* data extracted from nine subjects randomly extracted from the cursor scenario in dataset[45] (Tables 3 A, D, and Table 4 A), *(ii)* data extracted from nine subjects randomly extracted from the robot scenario in dataset[45] (Tables 3 B, E, and Table 4 B), *(iii)* 18 subjects, randomly selected extracting nine from the cursor scenario, and nine from the robot scenario in dataset[45] (Tables 3 C, F, and Table 4 C). Furthermore, testing is also performed considering three scenarios: *(i)* data extracted from all the six subjects from session 1 in dataset[18](Tables 3 A, B, C), *(ii)* data extracted from all the six subjects from session 2 in dataset[18](Tables 3 D, E, F), *(iii)* all the 12 subjects from both sessions in dataset[18] (Tables 4 A, B, C).

The evaluation process was repeated three times to perform cross-validation of our key results, which involve training and testing across different subjects, tasks, and experimental settings. To ensure independence across repetitions, we confirmed that the nine subjects selected for training were different in each run.

Tables 3 and 4 report only the best-performing classification model for each configuration, in order to highlight the effect of different training-testing combinations on generalization performance. Although multiple models were evaluated, the focus here is not on model comparison, but rather on the impact of training and testing configurations across datasets and tasks.

Each table also includes, at the top, a summary of the training and testing setup used to evaluate model performance. The values reported in the tables represent the mean and standard deviation across the three repetitions.

In Table 3 A, we show the performance of the model trained on the cursor session from dataset[45], tested with data from session 1 from the dataset[18]. Here, the **R1** requirement is met with the model built with all features, where accuracy is 78.0% and with the model built with 28 features, where recall is 98.8% and F1 is 87.7%. Moreover, the model built with 28 features have also the best time complexity score.

In Table 3 B, we continue showing the performance of the model trained on the robot session from dataset[45], tested with data from session 1 from dataset[18]. In this case, we have that the **R1** requirement is met by the model built with all the features, where accuracy is 79.0% and F1 is 88.3%, and also by the model built with

	Training set			**Testing set**		
	Number of subjects	**Scenario**		**Number of subjects**	**Scenario**	
	9+9+4+4	Cursor & Robot & Session 1 & Session 2		2+2+2+2	Cursor & Robot & Session 1 & Session 2	
	Dataset:[45] and[18] - Training: cursor, robot, session 1 and session 2 - Testing: cursor, robot, session 1 and session 2					
	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 features	ELM	No	4	**76.567%**	96.054%	**85.785%**
21 features	KNN	Yes	**2**	75.894%	**98.141%**	85.702%
28 features	SVM	No	**3**	75.534%	91.879%	84.684%
All features	KNN	Yes	**3**	74.382%	**99.087%**	85.062%
All samples	ELM	No	8	**89.402%**	96.059%	**92.647%**

Table 2. Results of cross-subject classification within datasets. The highest and the second-highest value for each metrics are highlighted in bold underlined and bold, respectively. The upper table (light blue) shows the organization of the training and testing sets.

	Training set			Testing set		
	Number of subjects	Scenario		Number of subjects	Scenario	
A	9	Cursor		6	Session 1	
B	9	Robot		6	Session 1	
C	9+9	Cursor & Robot		6	Session 1	
D	9	Cursor		6	Session 2	
E	9	Robot		6	Session 2	
F	9+9	Cursor & robot		6	Session 2	
	Dataset:[45] and[18] - Training: cursor[45] - Testing: session 1[18]					
A	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	Yes	**2**	73.222% ± 8.6%	90.131% ± 14.7%	83.743% ± 6.9%
21 Features	NB	Yes	**2**	73.211% ± 7.8%	97.871% ± 13.2%	87.326% ± 6.1%
28 Features	NB	Yes	**2**	**77.534% ± 1.6%**	**98.792% ± 2.0%**	**87.722% ± 1.1%**
All Features	NB	Yes	**5**	**78.047% ± 2.0%**	96.871% ± 3.6%	87.203% ± 1.4%
All Samples	NB	Yes	6	75.445% ± 4.3%	**98.594% ± 7.5%**	**87.578% ± 3.2%**
	Dataset:[45] and[18] - Training: robot - Testing: session 1					
B	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	No	**3**	78.208% ± 0.7%	99.063% ± 1.6%	87.754% ± 0.5%
21 Features	KNN	Yes	**2**	77.336% ± 1.2%	97.127% ± 1.7%	87.103% ± 0.8%
28 Features	KNN	Yes	**2**	76.471% ± 2.5%	95.315% ± 4.0%	86.433% ± 1.8%
All Features	KNN	Yes	**2**	**79.017% ± 0.1%**	**99.805% ± 0.2%**	**88.276% ± 0.1%**
All Samples	KNN	No	5	**78.800% ± 0.4%**	**99.904% ± 0.2%**	**88.137% ± 0.3%**
	Dataset:[45] and[18] - Training: cursor & robot[45] - Testing: session 1[18]					
C	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	KNN	Yes	**1**	72.495% ± 2.4%	87.342% ± 4.8%	83.307% ± 2.0%
21 Features	NB	No	5	70.951% ± 8.3%	85.584% ± 15.3%	81.814% ± 7.2%
28 Features	NB	Yes	**2**	**73.822% ± 8.9%**	**91.223% ± 15.2%**	**84.180% ± 7.0%**
All Features	KNN	Yes	3	**78.118% ± 0.7%**	**98.286% ± 1.4%**	**87.667% ± 0.5%**
All Samples	KNN	Yes	4	72.408% ± 9.5%	88.306% ± 17.1%	82.893% ± 8.0%
	Dataset:[45] and[18] - Training: cursor[45] - Testing: session 2[18]					
D	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	Yes	**3**	75.831% ± 6.0%	92.356% ± 9.9%	85.827% ± 4.4%
21 Features	NB	Yes	4	76.502% ± 5.2%	98.429% ± 9.0%	88.493% ± 3.8%
28 Features	NB	Yes	**2**	**79.174% ± 1.3%**	**98.915% ± 1.8%**	**88.709% ± 0.9%**
All Features	NB	Yes	**2**	**79.465% ± 1.7%**	96.783% ± 3.7%	88.122% ± 1.3%
All Samples	NB	Yes	6	78.182% ± 3.2%	**99.214% ± 5.3%**	**88.882% ± 2.2%**
	Dataset:[45] and[18] - Training: robot[45] - Testing: session 2[18]					
E	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	No	**3**	80.053% ± 0.4%	99.363% ± 1.1%	88.899% ± 0.3%
21 Features	NB	No	**2**	77.578% ± 4.7%	94.419% ± 9.7%	86.984% ± 3.6%
28 Features	KNN	Yes	**2**	77.988% ± 1.5%	96.716% ± 1.5%	87.514% ± 1.1%
All Features	KNN	Yes	5	**80.396% ± 0.3%**	**99.925% ± 0.1%**	**89.124% ± 0.2%**
All Samples	KNN	No	**2**	**80.285% ± 0.2%**	**99.825% ± 0.2%**	**89.060% ± 0.2%**
	Dataset:[45] and[18] - Training: cursor & robot[45] - Testing: session 2[18]					
F	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	SVM	Yes	4	**79.124% ± 1.7%**	**97.893% ± 2.6%**	**88.278% ± 1.1%**
21 Features	KNN	Yes	5	76.545% ± 0.2%	91.430% ± 0.4%	86.239% ± 0.1%
28 Features	NB	Yes	**2**	77.383% ± 5.3%	94.762% ± 9.1%	86.935% ± 3.9%
All Features	KNN	Yes	**3**	**79.684% ± 0.8%**	**98.578% ± 1.7%**	**88.633% ± 0.6%**
All Samples	KNN	Yes	4	72.087% ± 9.3%	85.855% ± 15.5%	82.702% ± 7.6%

Table 3. Tables from **A** to **F** report the results of cross-subject classification between datasets. The highest and the second-highest value for each metrics are highlighted in bold underlined and bold, respectively. The upper table (light blue) reports the organization of the training and testing sets.

	Training set			Testing set		
	Number of subjects	Scenario		Number of subjects	Scenario	
A	9	Cursor		6+6	Session 1 & session 2	
B	9	Robot		6+6	Session 1 & session 2	
C	9+9	Cursor & Robot		6+6	Session 1 & session 2	
	Dataset:[45] and[18] - Training: cursor[45] - Testing: session 1 & session 2[18]					
A	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	Yes	**2**	74.398% ± 7.0%	91.294% ± 12.1%	84.739% ± 5.4%
21 Features	NB	Yes	**4**	74.739% ± 6.2%	98.162% ± 11.0%	87.772% ± 4.8%
28 Features	NB	Yes	**4**	**78.209% ± 1.2%**	**98.856% ± 1.9%**	**88.074% ± 0.8%**
All Features	NB	Yes	**2**	**78.885% ± 1.2%**	96.783% ± 3.7%	**87.766% ± 0.9%**
All Samples	NB	Yes	6	76.339% ± 3.5%	**98.269% ± 6.1%**	87.762% ± 2.5%
	Dataset:[45] and[18] - Training: robot[45] - Testing: session 1 & session 2[18]					
B	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	NB	No	**3**	78.991% ± 0.8%	99.270% ± 1.3%	88.246% ± 0.5%
21 Features	KNN	Yes	**2**	77.831% ± 0.6%	97.478% ± 1.1%	87.576% ± 0.4%
28 Features	KNN	Yes	**2**	77.515% ± 1.5%	96.305% ± 2.5%	87.140% ± 1.0%
All Features	KNN	Yes	**2**	**79.555% ± 0.1%**	**99.826% ± 0.1%**	**88.566% ± 0.1%**
All Samples	KNN	Yes	**2**	**79.416% ± 0.1%**	**99.980% ± 0.2%**	**88.558% ± 0.1%**
	Dataset:[45] and[18] - Training: cursor & robot[45] - Testing: session 1 & session 2[18]					
C	Model	Oversampling	Time complexity score	Accuracy	Recall	F1
14 Features	SVM	Yes	**3**	**75.449% ± 3.6%**	**93.144% ± 6.1%**	**85.714% ± 2.5%**
21 Features	KNN	Yes	5	74.610% ± 0.9%	88.845% ± 2.9%	84.640% ± 0.9%
28 Features	SVM	No	**2**	73.973% ± 9.5%	84.761% ± 17.6%	81.733% ± 7.9%
All Features	KNN	Yes	**3**	**78.753% ± 0.8%**	**98.669% ± 1.6%**	**88.239% ± 0.6%**
All Samples	KNN	Yes	4	71.624% ± 8.9%	79.814% ± 15.7%	79.487% ± 7.5%

Table 4. Tables from **A** to **C** report the results of cross-subject classification between datasets. The highest and the second-highest value for each metrics are highlighted in bold underlined and bold, respectively. The upper table (light blue) reports the organization of the training and testing sets.

all the samples where recall is 99.9%. However, the model with 28 features achieves an increase for accuracy (+0.2%) and F1 (+0.1%), and a slightly decrease in recall (-0.1%) compared to the model with all signal samples. Moreover, the **R2** requirement is met using all the features, with the time complexity score equal to 2, while the time complexity score with all the samples is 5.

Table 3 C shows the performance achieved using, as testing scenario, the data from session 1 from[18], with the model trained on the scenario obtained merging cursor and robot sessions from[45]. In this case, the **R1** requirement is met by the model built with all the features, with accuracy of 78.1%, recall of 98.3% and F1 of 87.7%. Moreover, also the **R1** requirement is met by the same model, but the best time complexity score is about the model built with 28 features, that also for the **R1** requirement is a good candidate.

The same set of analyses was performed testing the approach with data from session 2 of dataset[18]. In particular, in Table 3 D, we use the model trained on the cursor session from dataset[45]. The **R1** requirement is partially met with all the features model, where accuracy is 79.5% and with all the samples model, where recall is 99.2% and F1 is 88.8. A good compromise is the model built with 28 features, where there is an increase in accuracy (+1.0%), and a slower decrease in Recall and F1 (-0.3% and -0.2%, respectively) in comparison with all the samples model and, moreover, the 28 features model has the best time complexity score.

Subsequently, Table 3 E reports the performance of the model trained on data from the robot session from dataset[45], tested with data from session 2 from dataset[18]. The **R1** requirement is met with the model built with all the features, with accuracy of 80.4%, recall of 99.9%, and F1 of 89.1%. However, this is the unique case where the best time complexity score is with all the samples model.

Table 3 F shows the results achieved training the model on data from the scenario obtained merging cursor and robot sessions from[45], and testing it on data from session 2 from dataset[18]. In this case, the **R1** requirement is met by the model built with all the features, with accuracy of 79.7%, recall of 98.6% and F1 of 88.6%. This model leads to achieving a large increase in accuracy (+7.6%), recall (+12.5%) and F1 (+0.1%) compared to the model built with all the samples. Additionally, this model returns the second-lowest time complexity score. We highlight that, in this case, the results obtained with 14, 21, 28, and all the features are better, regarding the **R1** and **R2** requirement, compared with the model built with all the samples.

In the end, in Tables 4 A, B, C, we consider, as testing part, the data from the scenario obtained merging session 1 and session 2 of the dataset[18]. In particular, in Table 4 A, we show the results using the model trained with data from the cursor scenario of dataset[45]. The **R1** requirement is partially met with the model built with all the features (accuracy of 78.9%) and the model built with 28 features (recall of 98.9% and F1 of 88.1%). The **R2** requirement is met with the model with all the features with a time complexity score of 2. Hence, the model

built with 28 the features has a better time complexity score, an increased accuracy (+1.9%), recall (+6.6%) and F1 (+0.3%), compared with the model built with all the samples.

In Table 4 B, the same analysis is performed, with the models trained with data from the robot scenario of dataset[45]. The results obtained for the model built with all the features are the best for accuracy (79.5%) and F1 score (88.6%). Instead for recall, the best model is all the samples (with 99.9%), but the recall from the model built with all the features is slightly smaller (-0.1%). Moreover, also the **R2** requirement is met for the all features model.

Finally,in Table 4 C, we show the results obtained from the model trained on data obtained merging the cursor and robot scenarios in dataset[45], and tested on data from the scenario obtained merging session 1 and session 2 of the dataset[18]. The **R1** and **R2** requirements are met with the feature-based models. The best accuracy (78.7%), recall (98.7%), F1 score (88.2%) and the second-lowest time complexity score are obtained with the all features model. This model returns a large increase in accuracy (+7.1%), recall (+18.9%) and F1 (8.7%), compared with the model built with all the samples. Moreover, in this case, all feature-based models are better than the model built with all the samples. Specifically, results comparable with the best are obtained from model built with only 14 features.

Comparison with the literature on cross-subject classification

In this paragraph, we compare the results presented in the previous section with those reported in the literature.

Table 5 reports a selection of studies[7,10,35–40] that performed cross-subject classification of ErrPs using different EEG datasets and evaluation metrics. Although the datasets and methodologies vary, this comparison provides a broader overview of existing approaches in the literature and helps contextualize the performance of our proposed method. To the best of our knowledge, no prior work has addressed *cross-subject classification* across different datasets. Instead, existing studies focus on cross-subject classification within a single experiment or task. For a more direct and fair comparison, we refer the reader to Table ?? in the Supplementary Materials, where we compare our method with other works that used the same dataset and implemented cross-subject

Study	Methodology	Model	Oversampling	Accuracy	Recall
Our	21 Features - Train and test on cursor, robot, session 1 and session 2[18,45]	KNN	Yes	75.894%	98.141%
	28 Features - Train cursor[45] - Test session 1[18]	NB	Yes	77.534%	98.792%
	All Features - Train robot[45] - Test session 1[18]	KNN	Yes	79.017%	**99.805%**
	All Features - Train cursor and robot[45] - Test session 1[18]	KNN	Yes	78.118%	98.286%
	28 Features - Train cursor[45] - Test session 2[18]	NB	Yes	79.174%	98.915%
	All Features - Train robot[45] - Test session 2[18]	KNN	Yes	**80.396%**	**99.925%**
	All Features - Train cursor and robot[45] - Test session 2[18]	KNN	Yes	79.684%	98.578%
	28 Features - Train cursor[45] - Test session 1 and session 2[18]	NB	Yes	78.209%	88.074%
	All Features - Train robot[45] - Test session 1 and session 2[18]	KNN	Yes	79.555%	**99.826%**
	All Features - Train cursor and robot[45] - Test session 1 and session 2[18]	KNN	Yes	78.753%	98.669%
[35] Gao et al.		DCNN +2		**87.940%**	
		LR		74.000%	
		HD		78.900%	
		RF		69.320%	
		SVM		71.000%	
		KNN		66.830%	
		DBN		72.840%	
		ShallowNet		79.860%	
[7] Pereira et al.		WRN-16-8		**80.510%**	41.230%
		WRN-16-1		73.870%	58.820%
[36] Zandi et al.	Zero crossing. similarity/dissimilarity index				88.340%
[37] Chu et al.	Fourier transform coefficient. PSD				86.670%
[38] Khan et al.	Wavelet transform coefficients. CNN				87.800%
[39] Truong et al.	STFT spectral images. CNN				89.800%
[40] Ruksar et al.	Statistical and morphological. MSPC				88.890%
[10] Wang et al.	Temporal: downsampling	LDA			68.72%
	Temporal: downsampling	PSVM			66.63%

Table 5. This table reports the metrics and the information that are available in the references regarding the problem of cross-subject classification of ErrPs. The highest and the second-highest value for each metrics are highlighted in bold underlined and bold, respectively. The accuracy achieved with our method is better than what is achieved by other works, except for[35] with the model DCNN +2. In particular, our models achieve the second best values: as regards accuracy, KNN model built with all samples, trained on robot data[45] and tested on session 2[18]; as regards recall KNN model using the robot scenario[45] as training and session 2[18] as testing.

classification. Due to the limited availability of such studies, this comparison is only possible for one of the two datasets used in our work.

Table 5 shows that the **R1** and **R2** requirements are met with the models built with the features, even though, regarding the **R1** requirement, the results achieved in[35] exhibit better accuracy. These discrepancies in the results are motivated by the differences in the classification methods, namely the machine learning algorithms, the time complexity, and the size of the dataset used for training. In particular, we used classic machine learning methods, while[35] used a convolution neural network. The time complexity of these algorithms is not reported in[35], but, often, the creation of a neural network requires more time than what is needed to train classic machine learning models, since neural networks are characterized by a larger number of parameters to optimize and more complex architectures. Moreover, they require much larger amounts of data for training the model[52].

Discussion and limitations

In this work, we consider the problem of implicit communication in a HRI context, to enhance the applicability of robotic solutions in diversified contexts.

While previous studies have proposed the use of ErrPs for error detection and correction in various contexts[19], our work takes a further step toward enabling real-time implicit communication by focusing on the generalization of ErrP classification across different subjects and experimental conditions. Although the present study does not implement a complete real-time application, the proposed feature-based, subject-independent ErrP classification represents a promising advancement toward practical EEG-based systems. Such models could be integrated into plug-and-play interfaces in industrial or assistive robotics, enabling real-time user error detection across diverse users without the need for retraining or calibration. We see this foundational contribution as a step toward the future deployment of robust, generalizable EEG-based implicit communication systems in real-world applications.

We propose an approach for the detection of ErrPs from different subjects performing different HRI tasks. We focus on cross-subject classification, thus avoiding subject-specific models that are impractical in the presence of different subjects. To evaluate the performance of the proposed detection methodology, we introduce two different requirements, referred to as **R1** and **R2**. The **R1** requirement refers to inclusion metrics such as accuracy, recall and F1, while the **R2** requirement regards the time to train the classification model (which is useful for allowing incremental learning in real time). We propose characterizing the EEG signal using a specific *features*, and show that this approach satisfies the considered requirements.

We start our analysis considering cross-subject classification *within datasets*, meaning that we merge the two datasets both in the training and testing steps. Here we obtain that the **R2** requirement is always met using the feature-based models, and the **R1** requirement is partially met using the all features model (in recall) and with the all signal samples model (in accuracy and F1). The best model among all the requirements is built with 21 features.

As a next step, we consider the cross-subject classification *between datasets*, meaning that we use different datasets in the training and testing steps. This allows to understand if, by combining different experimental setups, it is possible to create a generalized model to recognize ErrPs, usable in different contexts and by different people. In this way, the ErrP detection would be less specific in its use, and incremental learning would be easier, considering only one general, not subject-specific, classification model.

The sessions used in our study vary in experimental complexity. While the sessions from[18] share the same paradigm, the "robot" session from[45] involves more complex and semantically rich stimuli compared to the simpler "cursor" session. Interestingly, although[45] reports lower single-subject accuracy for the robot session due to higher variability, in our cross-subject experiments, training on the robot session sometimes led to better generalization. This suggests that the higher complexity may help the model learn more robust features that generalize better across subjects.

In general, our models built using features met the defined performance requirements.

Moreover, the models created using features have lower time complexity score: this implies that all the feature-based models are built in a shorter time compared with the all signal samples model.

We also compare the results obtained on cross-subject classification with the literature. We find that the **R1** and **R2** requirements are met with the models built with the features, even though, regarding the **R1** requirement, the work in[35] presents a better accuracy. However, it is worth remarking that a convolutional neural network is used in[35]. Although deep neural networks represent a promising avenue for EEG-based ErrP classification, their application remains limited by the relatively small size and heterogeneity of current datasets. Our goal is to build a generalizable feature-based model that can enable the future integration of multiple datasets, ultimately making it feasible to train robust deep learning architectures on larger, aggregated ErrP collection of data.

Furthermore, while previous works such as ERPENet[53] have explored multi-dataset generalization for ERP classification, specifically for the P300 component using deep neural networks, no prior study, to the best of our knowledge, has addressed the generalization of ErrP classification across subjects and tasks using feature-based machine learning approaches. Our work represents an initial step in this direction, laying the groundwork for future efforts aimed at combining multiple ErrP datasets to build a single, generalizable classification model.

To summarize the performance achieved with the proposed feature-based ErrP detection approach, considering all the analyses reported in the previous section, on average, the different groups of features return the results summarized in Table 6.

From these results we can see how the best compromise for satisfying the two requirements is using the model built with 28 features. Furthermore, it is worth noting that the **R2** requirement (time complexity score) is always better with any model built with the features: in fact, the results from the model built with all the samples are always worst. Hence, the model built with 28 features is the best one, because the performance are comparable with the model built with all the features but the time complexity score is better. Furthermore, we

Approach	Time complexity score	Accuracy	Recall	F1
14 features	**2.80**	76.419%	94.428%	86.278%
21 features	3.30	75.478%	94.372%	86.438%
28 features	**2.30**	**76.897%**	**95.072%**	**86.494%**
All features	3.00	**79.102**%	**98.392**%	**88.176%**
All samples	4.70	76.065%	94.418%	86.118%

Table 6. Average results for cross-subject classification. The highest and the second-highest value for each metrics for the proposed approach are highlighted in bold underlined and bold, respectively.

would like to point out that the best results are generally achieved exploiting the KNN or NB machine learning method, applying oversampling.

In general, using the features, we obtain better results for both the requirements. However, with reference to **R2**, it is worth remarking that the computation of the features, per se, introduces calculations that would otherwise not be needed. We consider 71 features, which are standard signal processing features, as described in the dedicated section in the *Methods*. Hence, with standard software libraries, the considered features can be efficiently and quickly computed. Also each considered machine learning model has a specific computational complexity, as reported in the literature. In particular, for all the considered models, the computational complexity ranges from $O(d)$, for DT[54], KNN[55], SVM[56], NB[57] and ELM[58], to $O(d^3)$ for LDA[59], where d is the number of elements used to train the models (in the considered cases, number of features or number of samples). Hence, the introduction of features significantly reduces d: with the considered datasets, with all the samples we obtain $d = 9666$ (358 samples for each epoch and 27 channels), instead, with all the features, we get $d = 1917$ (71 features for 27 channels). Furthermore, if we consider less features, the value of d further decreases. Achieving a reduction in d brings a great advantage, which is also reflected in the time complexity score, where the worst values are achieved with the model built with all the samples.

Based on these results, it is possible to conclude that the proposed classification methodology, based on the use of features, represents a powerful tool for the classification of the ErrP signals. These findings not only validate the method in terms of performance (R1) and computational efficiency (R2), but also support the core contribution of this work: the design and validation of a subject-independent, task-agnostic classification framework for ErrP detection, which does not rely on deep learning architectures. In contrast to previous approaches that often depend on subject-specific training or narrowly defined tasks, our feature-based strategy generalizes effectively across users and experimental setups while preserving model explainability and enabling incremental learning. This represents a significant step toward bridging the gap between controlled laboratory research and practical deployment of EEG-based systems, particularly in dynamic HRI scenarios, where different users interact with the same system over time and calibration is impractical. Therefore, the extensive statistical results presented in this work are not merely descriptive, but substantiate the robustness, scalability, and real-world readiness of the proposed approach, reinforcing the relevance and innovation of our contribution to the field.

Despite these promising results, some limitations must be acknowledged to better understand the applicability of our method in real-world contexts. First, EEG signals exhibit substantial variability across individuals due to physiological and cognitive differences, including factors such as fatigue, stress, and emotional state. Moreover, real-world environments often introduce additional noise and artifacts that are not present in controlled lab conditions. These factors may affect the robustness of the proposed method when deployed outside of standardized scenarios.

Nevertheless, the findings of this work represent a solid foundation for advancing toward generalizable and real-time ErrP detection. By demonstrating consistent performance across tasks and datasets without requiring subject-specific training, our feature-based approach offers a scalable solution suitable for dynamic HRI scenarios. Future research will focus on expanding the training collection of data, refining the feature selection under more diverse conditions, and exploring lightweight deep learning models as more data becomes available. We believe that this direction will ultimately enable robust, implicit brain-computer communication in practical and heterogeneous settings.

Methods

Datasets

In this section, we briefly present the datasets used in our analysis. The first dataset is described in detail in[45]. The dataset includes data from experiments with 11 subjects (six males and five females, average age 29.4 ± 7.4 years).

The dataset is divided into two different parts, called the cursor scenario and the robot scenario, respectively. In both scenarios, the performed experiments consist of an interaction task, where a square is presented to the participant, in the center of the monitor, and another square appears in three possible positions relative to the initial square (left, right, and up). This represents the initial stimulus, common for the two scenarios.

Then, in the cursor scenario, the subjects are requested to press the corresponding arrow on the keyboard to move the square from the center in the direction of the other square. Errors are introduced in the execution of these commands, with a probability between 20% and 50%, for each trial. If the square moves in the direction indicated by the subject, there is only an ERP signal; however, if the square moves in another direction, the subject perceives an error, and an ErrP is evoked.

The robot scenario is similar but, after the subject presses the arrow on the keyboard, the humanoid robot turns its head toward the direction indicated by the target (either correctly or incorrectly), providing a more

embodied representation of the action compared to the virtual square. Similarly, if the head of the robot moves in the correct direction, there is only an ERP signal, if the head of the robot moves in the wrong direction, there is an ErrP.

The stimuli are presented on an LCD computer monitor with a 60 Hz refresh rate. The experiment was programmed with Psychopy software[60]. They recorded 500 trials for participants with an average error rate of 35%.

The EEG signals are recorded with a Brain Products ActiChamp amplifier with 32 electrodes arranged according to international 10-20 systems (FP1, FP2, F3, F4, F7, F8, FC1, FC2, FC5, FC6, C3, C4, T7, T8, CP5, CP6, P3, P4, P7, P8, TP9, TP10, O1, O2, Fz, Cz, Pz, EOG1, EOG2, EOG3).

The second dataset used is available in[18]. This dataset records data from experiments performed by six subjects, in two different sessions, using 64 electrodes. Similar to the previous case, a square is presented in the middle of the screen and, after some time, another square (the target) appears on the right or on the left. During the experiment, the user has no control over the cursor's movement and is asked only to monitor the performance of the agent, knowing that the goal is to reach the target. The authors use a probability of error of 20% for session 1 and a probability of error of 40% for session 2.

This study is based on two publicly available EEG datasets, and no new experimental data were collected. As such, detailed information about the experimental environment (e.g., lighting conditions, ambient noise) and participants'internal states (e.g., fatigue or emotional status) is limited to what is reported by the original authors. Both datasets, however, were acquired under controlled laboratory settings, with consistent EEG acquisition protocols and structured task paradigms. For additional environmental or procedural details, we refer the reader to the original dataset documentation.

In our analyses, we use the pre-processing pipeline explained in[45] to analyze both datasets. First the EEG recording are filtered using a zero phase Hamming windowed sinc finite impulse response band-pass filter with cut-off frequencies 1 Hz and 20 Hz, in order to remove high frequency and power-line noise. Next, contaminated EEG channels are detected and interpolated based on kurtosis. Finally, the EEG signals are re-referenced to the common average reference (CAR) to further mitigate disturbances from external noise sources.

Although only two publicly available datasets were used in this study, each of them includes two distinct recording sessions conducted under different experimental conditions. This effectively results in four separate data collections, offering variation in cognitive tasks, acquisition settings, and subject behavior. Furthermore, the datasets include natural inter-subject variability in terms of age and gender, supporting the robustness of the validation. The primary objective of this study is to test the generalization of the proposed feature set across different users and sessions, rather than to optimize performance within a specific dataset. To this end, all sessions were combined and analyzed together, offering a meaningful testbed for assessing cross-subject classification. Future work will further extend this validation to include more diverse datasets and more complex task settings.

The features

List of features

All results in this paper are obtained by analyzing different feature groups and all the samples of the signals. Regarding the features, they were introduced in our previous work[44], where we conducted a statistical analysis on how significant they are in representing the ErrP signals. Therein, we considered several classes of features, namely temporal, frequency, and processing features. Such features were proposed in previous works on ErrP analysis, namely[10,40,61], to some extents, while we borrowed some other features from other signal processing domains, as discussed in[44].

The considered temporal features are the following: mean, prominence, maximum and minimum peaks (value and time), RMS, standard deviation, shape factor, crest factor, clearance factor, and impulse factor. The considered frequency features are: the maximum value and frequency of the Fourier transform, mean frequency, median frequency, band power, occupied bandwidth, power bandwidth, and peak location. The considered signal processing features are: Signal-to-Noise Ratio (SNR), the ratio of the amplitude of the first peak and the RMS value before the baseline (SNR1)[62], Total Harmonic Distortion (THD), signal to Noise and Distortion Ratio (SINAD). Instead, the selected statistical features analyzed are Kurtosis and Skewness. We also analyzed the Wavelet transform applied to two frequency bands[63]. We selected the [2-4] Hz band, which corresponds to δ rhythm (17 coefficients), and the [4-8] Hz band, which corresponds to θ rhythm of EEG signals (28 coefficients), as discussed in previous studies[64–66].

In summary, in[44] we identified a total of 71 features, composed of 12 temporal features, 8 frequency features, 4 processing features, 2 statistical features, 17 coefficients of Wavelet transform in the [2-4] Hz band and 28 coefficients of Wavelet transform in the [4-8] Hz band.

Selection of features

In[44], all the features were calculated on the dataset presented in[45] and explained in the *Datasets* section. Then, three different types of statistical analysis were performed: two non-parametric tests, namely Kolmogorov-Smirnov and Wilcoxon Rank tests, and one parametric test, namely t-test.

In studies[18,45], we assessed the significance of each feature using various statistical tests across both cursor and robot interaction paradigms, employing significance thresholds of 1%, 5%, and 10%. Building upon the findings of[18], we assigned weighted scores to each feature based on its significance level to rank feature relevance Specifically, we attribute the weights for every subject, every feature, and every statistical method, as follows:

- Weight = 3: significance level < 1%.
- Weight = 2: significance level < 5%.
- Weight = 1: significance level < 10%.

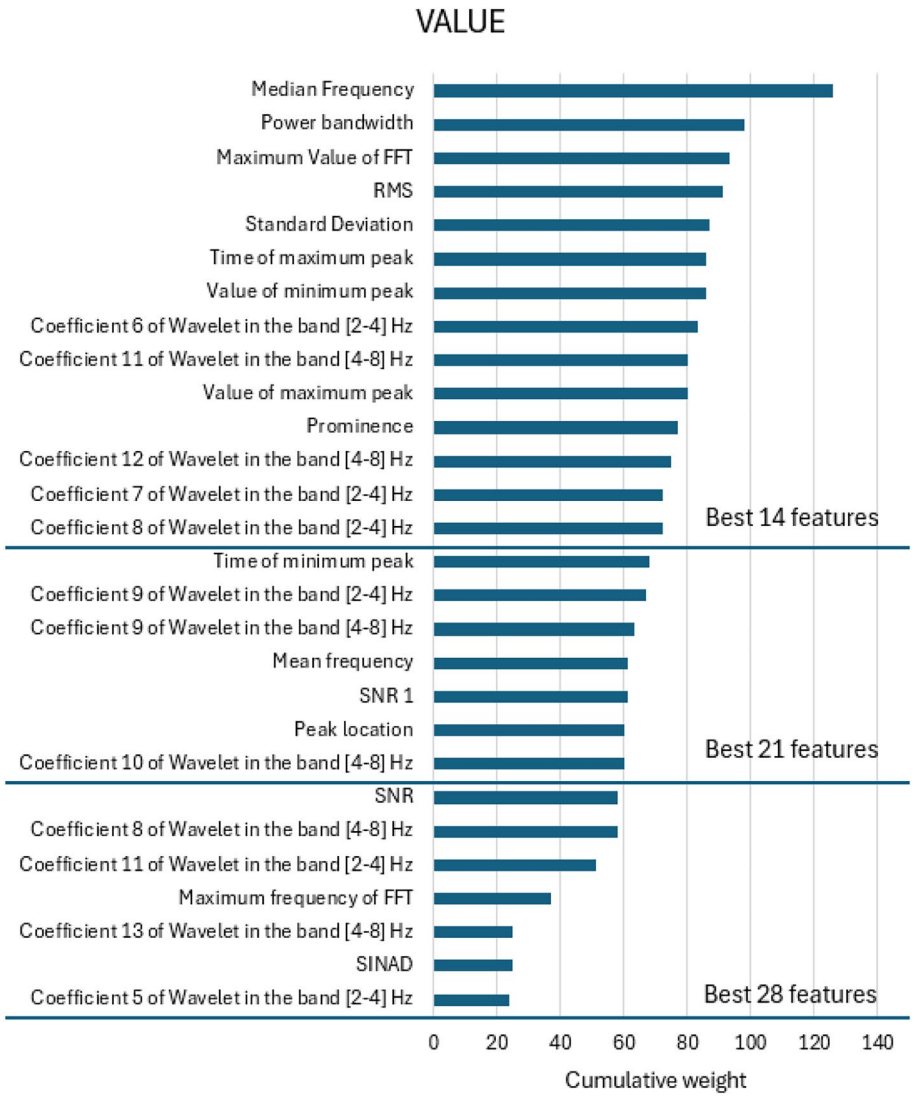

Fig. 2. Best 14, 21, and 28 features sorted by the significance level reported in[44], according to the analysis described in *Selection of features*.

- Weight = 0: significance level > 10 %.

Subsequently, we sum such weights to aggregate the results for each feature across all statistical methods, and sort them by the cumulative weight.

To identify the most relevant features, we define an index based on the weights attributed to each feature. The index is calculated by dividing the cumulative weight of each feature by the maximum possible weight. The maximum cumulative weight (equal to 198) is derived considering the case that a feature can detect ErrPs with the lowest significance level ($< 1\%$, corresponding to weight 3) for each of the 11 subjects, across the two scenarios, and for all the three statistical analyses ($3 \times 11 \times 2 \times 3 = 198$). Moreover, the best feature is the median frequency and its weight is 126 (63% of the maximum cumulative weight). Hence, we evaluate three groups of features: *(i)* features with a cumulative weight larger than or equal to 35% of the maximum (14 features), *(ii)* features with a cumulative weight larger than or equal to 30% of the maximum (21 features), and *(iii)* features with a cumulative weight larger than or equal to 10% of the maximum (28 features).

These thresholds were selected to balance feature relevance and model performance. Fig. 2 shows the ranking and corresponding weights of all features considered in the study, grouped according to these thresholds. The complete list of features is reported in the *List of features* section.

Unlike many recent studies, which often rely on PCA-based reduction or cross-validation-driven selection, our method is based on feature-level statistical significance, allowing us to retain interpretability and transparency in the selection process. For example,[67] extracted over 1500 features, mostly frequency-based, and used variance filters, correlation thresholds, and t-test voting to select 40 features. Similarly,[68] employed combinations of temporal, spectral, wavelet, and template-matching features, followed by PCA to reduce dimensionality while maintaining 95% variance. In[21], features from eight fronto-central channels were decorrelated via PCA and

selected using r-square scores, and[14] used fixed time windows to extract 280 pseudo-channel features for SVM classification, without statistical filtering.

In contrast, our ranking approach is statistically grounded and paradigm-agnostic, and provides a structured and explainable way to reduce feature count while preserving performance. As shown in Fig. 2, the most relevant features span across multiple domains, not limited to one category, confirming the advantage of a diverse and interpretable feature set for ErrP detection. Furthermore, the method demonstrated robustness across tasks and subjects, which supports its use in multi-scenario applications.

Classification models

We consider the following models:

- *Decision Tree (DT)*: this method makes a binary decision. The structure is composed of leaves, which are the classes, and branches, which are the conjunctions of features that lead to the classes. A high number of branches increases the computational complexity[48].
- *K-nearest neighbors (KNN)*: this method groups the features based on their distances. The class is assigned according to its surroundings, with a "majority vote". This method has a dependence on the number of features and is particularly useful with a low-dimensional feature vector[49].
- *Support vector machine* (SVM): this method builds a hyperplane that maximizes the distance between samples. It is preferred with a large number of samples[49].
- *Linear discriminant analysis (LDA)*: this method finds the linear combination to separate two different classes using a hyperplane. This technique has, in general, low computational complexity, but, if the dimensionality of the training sample is large, its computational time increases considerably[49].
- *Naive Bayes (NB)*: this method leverages the Bayes theorem and is based on probability density estimation of the data. Naive Bayes classifiers assign observations to the most probable class[50].
- *Ensemble learning method (ELM)*: this method combines different previous models to boost reliability. For this reason, it is known to produce reliable results[51].

For all classifiers, we used MATLAB's built-in automatic hyperparameter optimization function to select the most suitable configuration for each subject and fold. While hyperparameters were not manually fixed, we observed that the SVM classifier consistently selected a linear kernel function, and that for the KNN classifier, the optimal K value typically ranged between 15 and 40 across different runs. These values were associated with the best classification performance in our experimental setting.

The considered classification models were built using MATLAB software and two different laptops: Asus Intel Core i7 CPU@2.40 GHz and HP AMD Ryzen 7 7730U with Radeon Graphics CPU@2.00 GHz.

Metrics

In this work, we propose a set of requirements, in order to validate the performance of the proposed classification method. Such requirements are measured through the following metrics:

- *Accuracy*: it measures the ratio of correct predictions over the total number of instances evaluated. For the purpose of ErrP detection, this amounts to computing the ratio of ErrPs detected by a classification model over the number of epochs labeled with the occurrence of an ErrP.
- *Recall*, also known as true positive rate (TPR) or as sensitivity: it is used to measure the ratio of EEG epochs with an ErrP that are correctly classified. This parameter is the main metrics used in the detection of P300[41].
- *F1*: it is the harmonic mean of precision and recall, it is an important parameter for unbalanced dataset[42]. Precision represents the proportion of correctly predicted positive results out of all instances predicted as positive, instead recall measures proportion of all actual positives that were classified correctly as positives.
- *Time to build the model*: it is calculated as the difference between the start and end times of training. In the results of this work, we evaluate the time to build the model as *Time complexity score* (introduced in Table 1). This quantity is important when training needs to be frequently repeated over time. For instance, this can happen in an incremental learning situation, where the model is initially created with some data and, later, more data are added, allowing to incrementally update the initial model.

Data availability

All data analysed during this study are included in these published articles: Ehrlich, S. & Cheng, G. A feasibility study for validating robot actions using eeg-based error-related potentials. Int. J. Soc. Robotics 11, DOI: 10.1007/s12369-018-0501-8 (2019). Chavarriaga, R. & Millán, J. d. R. Learning from eeg error-related potentials in non-invasive brain-computer interfaces. IEEE transactions on neural systems rehabilitation engineering 18, 381-388 (2010).

Received: 4 April 2025; Accepted: 5 September 2025
Published online: 08 October 2025

References

1. Antonelli, M. G., Beomonte Zobel, P., Manes, C., Mattei, E. & Stampone, N. Emotional intelligence for the decision-making process of trajectories in collaborative robotics. *Machines* **12**, 113 (2024).
2. Lagomarsino, M., Lorenzini, M., Momi, E. & Ajoudani, A. An online framework for cognitive load assessment in industrial tasks. *Robotics Computer-Integrated Manuf.* **78**, 102380 (2022).
3. Lorenzini, M., Lagomarsino, M., Fortini, L., Gholami, S. & Ajoudani, A. Ergonomic human-robot collaboration in industry: A review. *Front. Robotics AI* **9**, 813907 (2023).

4. Mattei, E. et al. Deep learning architecture analysis for eeg-based bci classification under motor execution. In *2024 IEEE 37th International Symposium on Computer-Based Medical Systems (CBMS)* 549–555 (IEEE, 2024).
5. Arpaia, P., Moccaldi, N., Prevete, R., Sannino, I. & Tedesco, A. A wearable eeg instrument for real-time frontal asymmetry monitoring in worker stress analysis. *IEEE Trans. Instrum. Measurement* **69**, 8335–8343 (2020).
6. Tang, Z. et al. Hand rehabilitation exoskeleton system based on eeg spatiotemporal characteristics. *Expert Syst. Appl.* **270**, 126574 (2025).
7. Pereira, A., Padden, D., Jantz, J., Lin, K. & Alcaide-Aguirre, R. Cross-subject eeg event-related potential classification for brain-computer interfaces using residual networks. Frontiers in Computational Neuroscience. https://doi.org/10.13140/RG.2.2.16257.10086 (2018).
8. Kulic, D. & Croft, E. A. Affective state estimation for human-robot interaction. *IEEE Transa. Robotics* **23**, 991–1000 (2007).
9. Parra, L., Spence, C., Gerson, A. & Sajda, P. Response error correction - a demonstration of improved human-machine performance using real-time eeg monitoring. *Neural Syst. Rehabilitation Eng. IEEE Transa.* **11**, 173–177. https://doi.org/10.1109/TNSRE.2003.814446 (2003).
10. Wang, S., Lin, C.-J., Wu, C. & Chaovalitwongse, W. Early detection of numerical typing errors using data mining techniques. *IEEE Transa. Syst. Man Cybernetics, Part A* **41**, 1199–1212. https://doi.org/10.1109/TSMCA.2011.2116006 (2011).
11. Lotte, F. et al. A review of classification algorithms for eeg-based brain-computer interfaces: A 10-year update. *J. Neural Eng.* https://doi.org/10.1088/1741-2552/aab2f2 (2018).
12. Swamy Bellary, S. A. & Conrad, J. M. Classification of error related potentials using convolutional neural networks. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* 245–249, https://doi.org/10.1109/CONFLUENCE.2019.8776901 (2019).
13. Kumar, A., Pirogova, E. & Fang, J. Classification of error-related potentials using linear discriminant analysis. In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)* 18–21, https://doi.org/10.1109/IECBES.2018.8626709 (2018).
14. Kim, S.-K., Kirchner, E., Stefes, A. & Kirchner, F. Intrinsic interactive reinforcement learning - using error-related potentials for real world human-robot interaction. *Sci. Rep.* https://doi.org/10.1038/s41598-017-17682-7 (2017).
15. Sur, S. & Sinha, V. Event-related potential: An overview. *Ind. Psychiatry J.* **18**, 70–3. https://doi.org/10.4103/0972-6748.57865 (2009).
16. Holroyd, C. B. & Coles, M. G. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychol. Rev.* **109**, 679 (2002).
17. Falkenstein, M., Hoormann, J., Christ, S. & Hohnsbein, J. Erp components on reaction errors and their functional significance: A tutorial. *Biol. Psychol.* **51**, 87–107. https://doi.org/10.1016/S0301-0511(99)00031-9 (2000).
18. Chavarriaga, R. & Millán, J. d. R. Learning from eeg error-related potentials in noninvasive brain-computer interfaces. IEEE transactions on neural systems and rehabilitation engineering **18**, 381–388 (2010).
19. Chavarriaga, R., Sobolewski, A. & Millan, J. d. R. Errare machinale est: The use of error-related potentials in brain-machine interfaces. Frontiers in Neuroscience **8**, https://doi.org/10.3389/fnins.2014.00208 (2014).
20. Ciabattoni, L. et al. Errp signals detection for safe navigation of a smart wheelchair. In *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)* 269–272, https://doi.org/10.1109/ISCE.2019.8900993 (2019).
21. Iturrate, I., Chavarriaga, R., Montesano, L., Minguez, J. & del R. Millán, J. Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control. Sci. Rep. **5** (2015).
22. Liu, D., Chen, W., Pei, Z. & Wang, J. A brain-controlled lower-limb exoskeleton for human gait training. *Rev. Sci. Instrum.* **88**(10), 104302 (2017).
23. Accogli, A. et al. EMG-Based Detection of User's Intentions for Human-Machine Shared Control of an Assistive Upper-Limb Exoskeleton. *Spinger* **16**, 181–185 (2017).
24. Spüler, M. & Niethammer, C. Error-related potentials during continuous feedback: using eeg to detect errors of different type and severity. Front. Human Neurosci. **9** (2015).
25. Jeong, J.-H., Kim, K.-T., Yun, Y.-D. & Lee, S.-W. Design of a brain-controlled robot arm system based on upper-limb movement imagery. In *2018 6th International Conference on Brain-Computer Interface (BCI)*, 1–3 (IEEE, 2018).
26. Lin, C.-T. et al. A wireless multifunctional ssvep-based brain-computer interface assistive system. *IEEE Transa. Cognitive and Develop. Syst.* **11**, 375–383 (2018).
27. Nurseitov, D., Serekov, A., Shintemirov, A. & Abibullaev, B. Design and evaluation of a p300-erp based bci system for real-time control of a mobile robot. In *2017 5th International Winter Conference on Brain-Computer Interface (BCI)*, 115–120 (IEEE, 2017).
28. Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G. & Vaughan, T. M. Brain-computer interfaces for communication and control. *Clinical Neurophys.* **113**, 767–791 (2002).
29. Hwang, H.-J., Kim, S., Choi, S. & Im, C.-H. Eeg-based brain-computer interfaces: a thorough literature survey. *Int. J. Human-Computer Interaction* **29**, 814–826 (2013).
30. Salazar-Gomez, A. F., DelPreto, J., Gil, S., Guenther, F. H. & Rus, D. Correcting robot mistakes in real time using eeg signals. In *2017 IEEE international conference on robotics and automation (ICRA)*, 6570–6577 (IEEE, 2017).
31. Ehrlich, S. K. & Cheng, G. Human-agent co-adaptation using error-related potentials. *J. neural Eng.* **15**, 066014 (2018).
32. Lopes-Dias, C., Sburlea, A. I. & Müller-Putz, G. R. Online asynchronous decoding of error-related potentials during the continuous control of a robot. *Sci. Rep.* **9**, 17596 (2019).
33. Ball, T., Kern, M., Mutschler, I., Aertsen, A. & Schulze-Bonhage, A. Signal quality of simultaneously recorded invasive and non-invasive eeg. *NeuroImage* **46**, 708–716. https://doi.org/10.1016/j.neuroimage.2009.02.028 (2009).
34. Urigüen, J. A. & Garcia-Zapirain, B. Eeg artifact removal—state-of-the-art and guidelines. *J. Neural Eng.* **12**, 031001 (2015).
35. Gao, C., Li, Z., Ora, H. & Miyake, Y. Improving error related potential classification by using generative adversarial networks and deep convolutional neural networks. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2468–2476, https://doi.org/10.1109/BIBM49941.2020.9313238 (2020).
36. Shahidi Zandi, A., Tafreshi, R., Javidan, M. & Dumont, G. A. Predicting epileptic seizures in scalp eeg based on a variational bayesian gaussian mixture model of zero-crossing intervals. *IEEE Transa. Biomed. Eng.* **60**, 1401–1413. https://doi.org/10.1109/TBME.2012.2237399 (2013).
37. Chu, H., Chung, C. K., Jeong, W. & Cho, K.-H. Predicting epileptic seizures from scalp eeg based on attractor state analysis. *Comput. Methods Programs Biomed.* **143**, 75–87. https://doi.org/10.1016/j.cmpb.2017.03.002 (2017).
38. Khan, H., Marcuse, L., Fields, M., Swann, K. & Yener, B. Focal onset seizure prediction using convolutional networks. *IEEE Transa. Biomed. Eng.* **65**, 2109–2118. https://doi.org/10.1109/TBME.2017.2785401 (2018).
39. Truong, N. D. et al. A generalised seizure prediction with convolutional neural networks for intracranial and scalp electroencephalogram data analysis. arXiv preprint arXiv:1707.01976 (2017).
40. Rukhsar, S., Khan, Y., Farooq, O., Sarfraz, M. & Khan, A. Patient-specific epileptic seizure prediction in long-term scalp eeg signal using multivariate statistical process control. *Irbm* **40**, 320–331 (2019).
41. Cecotti, H. & Graeser, A. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *IEEE Transa. Pattern Anal. Mach. Intell.* **33**, 433–45. https://doi.org/10.1109/TPAMI.2010.125 (2011).
42. Abdelhamid, M. & Desai, A. Balancing the scales: A comprehensive study on tackling class imbalance in binary classification. arXiv preprint arXiv:2409.19751 (2024).
43. He, H. & Garcia, E. A. Learning from imbalanced data. *IEEE Trans. Knowledge Data Eng.* **21**, 1263–1284. https://doi.org/10.1109/TKDE.2008.239 (2009).

44. Fava, A., Villani, V. & Sabattini, L. Exploring the most significant features for eeg errp detection through statistical analysis. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)* (2024).
45. Ehrlich, S. & Cheng, G. A feasibility study for validating robot actions using eeg-based error-related potentials. *Int. J. Soc. Robotics* https://doi.org/10.1007/s12369-018-0501-8 (2019).
46. Mohammed, R., Rawashdeh, J. & Abdullah, M. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*, 243–248 (IEEE, 2020).
47. Hossin, M. & Sulaiman, M. N. A review on evaluation metrics for data classification evaluations. *Int. J. Data Mining Knowledge Manag. Process* **5**, 1 (2015).
48. Loh, W.-Y. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica* **12**, 361–386 (2002).
49. Rashid, M. et al. Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review. *Front. Neurorobotics* https://doi.org/10.3389/fnbot.2020.00025 (2020).
50. Hastie, T., Tibshirani, R. & Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) (Springer, 2009).
51. Freund, Y. A more robust boosting algorithm. arXiv preprint arXiv:0905.2138 (2009).
52. Goodfellow, I. Deep learning (2016).
53. Ditthapron, A., Banluesombatkul, N., Ketrat, S., Chuangsuwanich, E. & Wilaiprasitporn, T. Universal joint feature extraction for p300 eeg classification using multi-task autoencoder. *IEEE access* **7**, 68415–68428 (2019).
54. QUINLAN, J. Induction of decision trees. Machine Learning (1986).
55. Adamczyk, J. k nearest neighbors computational complexity. Towards Data Science (2020).
56. Wang, H., Zhu, Z. & Shao, Y. Fast support vector machine with low-computational complexity for large-scale classification. *IEEE Trans. Syst. Man Cybernetics: Syst.* **54**, 4151–4163. https://doi.org/10.1109/TSMC.2024.3375021 (2024).
57. Fleizach, C. A naive bayes classifier on 1998 kdd cup (2006).
58. BREIMAN, L. Random forests. Machine Learning https://doi.org/10.1023/A:1010933404324 (2001).
59. Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).
60. Peirce, J. et al. Psychopy2: Experiments in behavior made easy. Behavior Res. Methods **51**, https://doi.org/10.3758/s13428-018-01193-y (2019).
61. Ventouras, E. M., Asvestas, P., Karanasiou, I. & Matsopoulos, G. K. Classification of error-related negativity (ern) and positivity (pe) potentials using knn and support vector machines. *Comput. Biol. Med.* **41**, 98–109. https://doi.org/10.1016/j.compbiomed.2010.12.004 (2011).
62. Shin, J. et al. Wearable eeg electronics for a brain-ai closed-loop system to enhance autonomous machine decision-making. *npj Flexible Electron.* **6**, 32. https://doi.org/10.1038/s41528-022-00164-w (2022).
63. Mallat, S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 674–693 (1989).
64. Ferracuti, F. et al. A human-in-the-loop approach for enhancing mobile robot navigation in presence of obstacles not detected by the sensory set. Front. Robotics AI **9**, https://doi.org/10.3389/frobt.2022.909971 (2022).
65. Omedes, J., Iturrate, I., Chavarriaga, R. & Montesano, L. Asynchronous decoding of error potentials during the monitoring of a reaching task. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 3116–3121, https://doi.org/10.1109/SMC.2015.541 (2015).
66. Sanei, S. & Chambers, J. A. EEG signal processing and machine learning (John Wiley & Sons, 2021).
67. Dias, C. et al. Classification of erroneous actions using eeg frequency features: implications for bci performance. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* 629–632 (IEEE, 2021).
68. Usama, N., Kunz Leerskov, K., Niazi, I. K., Dremstrup, K. & Jochumsen, M. Classification of error-related potentials from single-trial eeg in association with executed and imagined movements: A feature and classifier investigation. Med. Biol. Eng. Comput. **58**, 2699–2710 (2020).

Author contributions

AF, VF, L.S. conceived the analyses, AF conducted the analyses, AF analyzed the results, V.V., L.S. supervised the work, A.F. wrote original draft. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-19172-7.

Correspondence and requests for materials should be addressed to A.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.