Aalborg Universitet

Automated labeling and online evaluation for self-paced movement detection BCI

Zhang, Dalin; Hansen, Christoffer; De Frène, Fredrik; Kærgaard, Simon Park; Qian, Weizhu; Chen, Kaixuan

Link to publication from Aalborg University

Automated labeling and online evaluation for self-paced movement detection BCI

Dalin Zhang, Christoffer Hansen, Fredrik De Frène, Simon Park Kærgaard, Weizhu Qian, Kaixuan Chen *

Department of Computer Science, Aalborg University, Aalborg Øst, 9220, Denmark

ARTICLE INFO

ABSTRACT

Electroencephalogram (EEG)-based brain–computer interfaces (BCIs) allow users to use brain signals to control external instruments, and movement intention detecting BCIs can aid in the rehabilitation of patients who have lost motor function. Existing studies in this area mostly rely on cue-based data collection that facilitates sample labeling but introduces noise from cue stimuli; moreover, it requires extensive user training, and cannot reflect real usage scenarios. In contrast, self-paced BCIs can overcome the limitations of the cue-based approach by supporting users to perform movements at their own initiative and pace, but they fall short in labeling. Therefore, in this study, we proposed an automated labeling approach that can cross-reference electromyography (EMG) signals for EEG labeling with zero human effort. Furthermore, considering that only a few studies have focused on evaluating BCI systems for online use and most of them do not report details of the online systems, we developed and present in detail a pseudo-online evaluation suite to facilitate online BCI research. We collected self-paced movement EEG data from 10 participants performing opening and closing hand movements for training and evaluation. The results show that the automated labeling method can contend well with noisy data compared with the baseline labeling method. We also explored popular machine learning models for online self-paced movement detection. The results demonstrate the capability of our online pipeline, and that a well-performing offline model does not necessarily translate to a well-performing online model owing to the specific settings of an online BCI system. Our proposed automated labeling method, online evaluation suite, and dataset take a concrete step towards real-world self-paced BCI systems.

1. Introduction

A brain–computer interface (BCI) aims to translate brain activities into commands to communicate with or control external devices. Brain activity can be reflected by voltage fluctuations from the scalp and obtained from electrodes placed on the scalp as an electroencephalogram (EEG). Among the various types of BCIs, the EEG-based BCI is one of the most attractive because of its zero clinical risk, portability, and cheap instrumentation [1,2]. The use of EEG-based BCIs to detect a participant's motor intention has been intensively studied to help patients suffering from movement impairments because of neurological injuries or diseases to restore their motor capacity [3–5].

Machine learning models are key components of many biomedical and health applications [6–8] including EEG-based BCIs [9–11]. To obtain a reliable machine learning model, it is mandatory to train the model with massive EEG data and corresponding labels. The most widely used paradigm for collecting EEG data is cue-based because of its simple labeling procedure. Specifically, participants followed a timed auditory or visual cue from an external device to execute movement intention [12,13]. This makes it easy to detect the time points that signify the movement intention, labeling the EEG data accordingly. However, it may also lead to associations between afferent stimuli from timed cues and motor cortical activities [14], which do not map well to real-world BCIs where patients do not follow visual or auditory cues to initiate movement intention. Furthermore, patients with cognitive and perceptual deficits that impede their ability to react to cues may have difficulty using cue-based methods [14], which require extensive training of patients to respond to cues.

In contrast, the self-paced approach allows participants to perform movements at their own initiative and pace to closely approximate real-world scenarios. The movement-related cortical potential (MRCP) is the most promising cortical paradigm used in self-paced approaches [1], which naturally occurs in EEG signals approximately 2 s before the point of movement intention or execution [15]. The MRCPs are elicited in the same way for healthy

* Corresponding author.
E-mail address: kchen@cs.aau.dk (K. Chen).

and non-healthy participants and for movement execution and intention. These characteristics simplify the EEG collection process by using the EEG signals for movement execution from healthy participants. In this study, we focused on using MRCPs for self-paced EEG movement execution detection.

Considering this, electromyography (EMG) signals are commonly recorded along with EEG signals from participants to serve as timing cues for marking movements in EEG signals. EMG signals were recorded from skeletal muscle activities using non-invasive electrodes placed on top of the activated muscle on a participant's skin during movement execution. The resulting signals capture the muscle activity during the participant's movements. These signals can then be cross-referenced with the EEG signals to distinguish between rest and movement in EEG signals, and thus provide labels for training machine learning models. The process is currently performed manually, making it prone to human errors and difficult to apply to large volumes of data [16]. This shortcoming severely limits the extensive research and development of the self-paced approach.

Moreover, most existing EEG studies are based on offline evaluations, where the results are summarized based on discrete EEG segments. This is different from real-world/online scenarios, where a BCI instantaneously receives continuous EEG signals and translates them into commands instantaneously. A common scheme of a reliable online BCI [16] is to first slice the continued EEG signals into discrete segments, obtain the detection results from the discrete EEG segments, and make the final decision based on a series of consecutive detection results. For example, if n consecutive detection results of discrete segments indicate movements, where n is an empirically determined parameter, the final decision would be recognized as movement; otherwise, it would be recognized as rest. This type of scheme is intended to achieve a low false positive (FP) rate because in practical applications, false positive detection can lead to undesired movements, which in turn may cause unwanted and potentially dangerous situations. In addition, online BCIs typically have a higher tolerance for error detection. For instance, a slightly later movement detection (c.f. onset) is also acceptable and should be considered a successful detection because the cortical potentials of brain activity vary in duration. Therefore, an offline evaluation does not necessarily reflect the real performance or usability in real, online use cases. However, research on online BCIs is rare owing to their complex settings and the lack of a common pipeline; moreover, many current studies with online exploration do not offer concrete details [17].

Considering the above limitations of the current research, we propose an automated labeling method for self-paced EEG movement detection and design a generic pipeline for pseudo-online evaluation. The automated labeling approach is a threshold-based clustering algorithm that first recognizes EMG signals of movement via dynamic thresholds, and then clusters these signals according to their temporal proximity. Finally, close clusters are merged and outliers are removed to refine the clustering results. The process is repeated to achieve satisfactory results. The resulting movement clusters are used to cross-reference the movement onsets of the recorded EEG signals, thus enabling automated EEG sample labeling. We collect EEG data of self-paced movement from 10 healthy participants and label samples with the proposed automated labeling method. The quantitative and visualization results demonstrate the effectiveness of the proposed automated labeling approach. Then, we present a detailed pseudo-online evaluation pipeline that simulates a real online BCI system with the techniques of preprocessing buffer, dwell heuristic, and freezing procedure for system robustness. We explore popular machine learning models for self-paced movement detection using our pseudo-online evaluation pipeline. The experimental results

demonstrate the capabilities of our pipeline and indicate that offline evaluation conclusions do not necessarily reflect online results, especially considering the detection latency of online BCI.

The main contributions of this paper are summarized as follows:

- We have made our self-paced movement detection dataset publicly available[1] for future research in this community. It contains simultaneously recorded EEG, EMG, and electrooculography (EOG) data from 10 healthy participants as they perform self-paced hand opening and closing movements by mimicking the control of a robotic soft glove. Rather than following a specified cue in most existing cue-based datasets, the participants move at their own pace and will.

- We have proposed an automated labeling method for self-paced movement EEG signals. Instead of using a manual threshold, we have designed a threshold-based clustering algorithm to detect and cluster movement within EMG signals. The detected movement clusters have been cross-referenced to EEG signals for movement onset labeling. This can address the bottleneck of expanding the self-paced movement detection research.

- We have attempted to fill the gap in online BCI research by developing a pseudo-online evaluation pipeline, which will simulate real BCI systems and facilitate online BCI research. We have further investigated different machine learning models for online movement detection to validate the proposed pipeline and present the difference between the online and offline results.

Section 2 describes the details of the equipment setup and data collection process; Section 3 presents the automated EEG labeling approach; Section 4 introduces the online evaluation suite; Section 5 reports the experiment results; Section 6 covers discussion of the findings of this paper; finally, Section 7 concludes the whole paper.

2. Data collection

This section describes the setup and procedure of our self-paced EEG data collection. Fig. 1 gives an overview of the key components of the procedure.

2.1. Instrumentation setup

We used the g.GAMMAsys system (g.tec GmbH, Austria) to collect EEG data. It has a g.GAMMAcap placed on the participant's head (see Fig. 1(A) rest position), which is connected to a g.USBamp amplifier via a g.GAMMAbox. The g.GAMMAcap has nine EEG electrodes – T7, C5, C3, C1, Cz, C2, C4, C6, and T8 – following the international standard 10–20 system (see Fig. 1(B)). The ground electrode was placed at AFz, and the reference electrode was placed on the left earlobe at A1. We used Fp1 to record EOG signals to detect eye-movement artifacts. EMG signals were recorded alongside the EEG signals using an EMG electrode placed on the forearm with a reference electrode (cyan) on the wrist and a ground electrode (red) on the lateral epicondyle of the elbow of the participant's dominant arm, as shown in Fig. 1(B). The EMG electrodes were also connected to the g.USBamp amplifier. We selected the EMG electrode placements as they can well capture the movement of the activated muscles when performing the movement tasks [16]. All EEG and EMG electrodes were filled with conductive gel to ensure good contact between the skin and the electrodes. We recorded all signals with a sampling rate of 1200 Hz without using the embedded filtering of the amplifier.
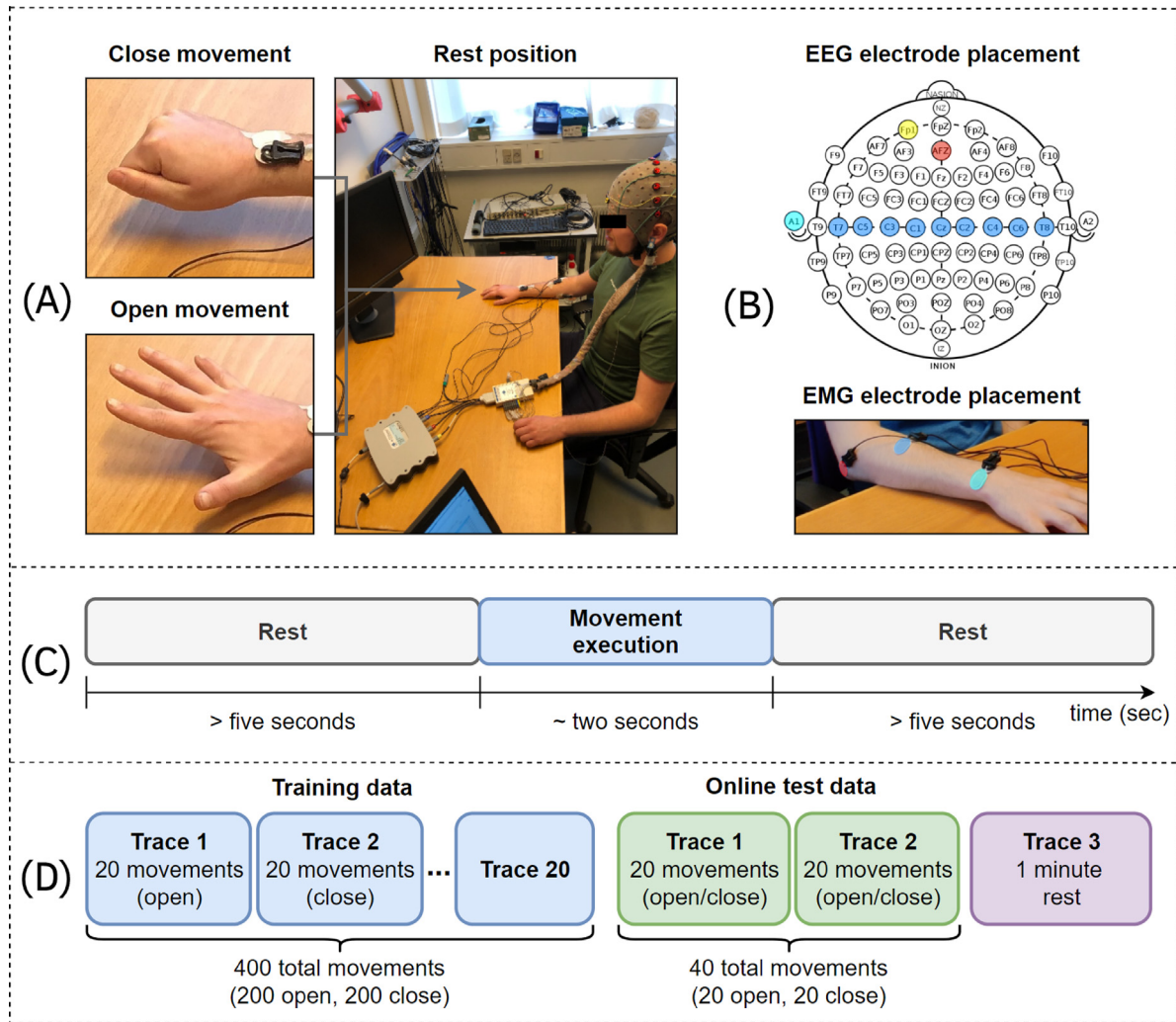
---

[1] https://github.com/P9-MI-BCI/mind-reading-and-control

Fig. 1. Overview of the experimental setup for data collection. (A) Movement (open and close) and reset setup. (B) EEG and EMG electrode placement. EEG electrode placement follows the international standard 10–20 system. The nine used EEG electrodes are marked with blue, the reference electrode with cyan, the EOG electrode with yellow, and the ground electrode with red. EMG electrode layout is shown with the ground electrode on the elbow, the main electrode on the forearm, and the reference electrode on the wrist, following the same color scheme. (C) Timeline for the execution of a single movement during data collection. (D) Illustration of training and test traces. We record 20 traces of training data and 2 traces of online test data for each subject resulting in 400 movements and 40 movements, respectively. We additionally record one trace of rest.

We recorded signals from the opposite hemisphere of the dominant part of the motor cortex by placing electrodes on either side of the longitudinal fissure. The symmetrical electrode placement along the mid-scalp allowed us to record from both left- and right-handed participants without changing the layout.

2.2. Participants

We recorded data from 10 participants, consisting of eight males and two females, with an average age of 24 ± 1.25. Nine participants were right-handed and one participant was left-handed. All participants were healthy and without any known neurological disorders. None of the participants were acquainted with BCI systems or had any prior experience with the data collection procedure. All participants agreed to the collection of data.

2.3. Data recording

Before data recording, the participants were instructed to comfortably sit in a chair in front of a table with the dominant arm resting on the table (see Fig. 1(A)). Then, we instructed the

subjects to execute movement tasks. During data recording, the participants executed movements in a self-paced manner without excess external communication, and all instructions were given prior to the recording.

For the recording of training data, a participant executes the movement (either opening or closing a hand) for approximately 2 s and rests for at least 5 s each movement (see Fig. 1(C)). Participants do not consciously count seconds or focus on maintaining a constant pace. Conversely, we monitor the speed and number of movements of a participant, and subsequently indicate to the participant when the halfway mark is reached and when a trace is complete. We also instructed the participants not to blink, swallow, or exercise other facial muscles to minimize artifacts within the EEG signals while executing the movements. We asked the participants to perform such motions while being in the rest position between movement task executions. They were allowed to hold a small break of 1–5 min after each trace. An extended break was provided after 12 traces before continuing with the rest of the traces. From each participant, we recorded 20 *traces*, each of which consisted of 20 movements of a single type, either closing or opening of a hand (see Fig. 1(D), Training data). Thus, each participant executed 200 opening and closing movements,
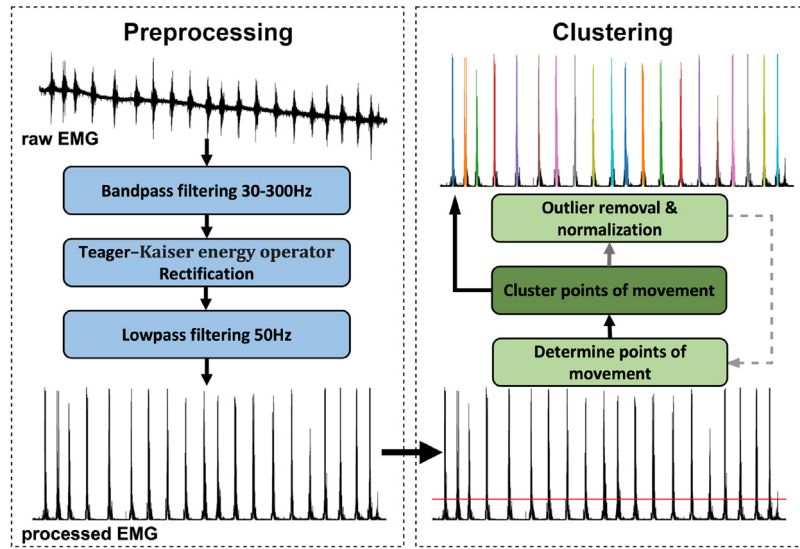
Fig. 2. Movement detection from EMG: preprocessing and clustering of the EMG signals of movement. (1) First, the raw EMG signal is bandpass filtered at 30–300 Hz using a sixth-order Butterworth filter. Then, the signal is rectified with TKEO and lowpass filtered at 50 Hz before being passed on to the clustering algorithm. (2) The clustering algorithm calculates a threshold to recognize movement signal points from EMG, which are clustered based on proximity. We then remove outlier clusters and perform normalization. This process, denoted by the gray dashed arrows, iterates until the expected number of clusters is identified.

respectively, for training data recording. Moreover, we recorded 1 min of rest, during which subjects did not execute any movements. For the recording of test data, participants alternately executed the opening and closing movements of hand in a trace. The other instructions were the same as those for the training data recording. Each subject recorded two test traces consisting of 40 movements in total (see Fig. 1(D), Online test data).

In summary, the dataset includes three classes: opening a hand, closing a hand, and rest. In this study, we considered the opening and closing of hand as a single movement class and performed binary classification of the brain-switch case [15,16] that reflects the opposing states of an external device.

3. Automated labeling

In the cue-based approach, data are labeled at the cue's position regardless of whether the participant has executed the movement at the specified timed cue, leading to the misalignment of the movement in the training samples. Our self-paced method records EMG signals from the skeletal muscle activity along with EEG signals, which allows us to detect when the movements are actually executed and align them with the EEG signals. Therefore, in this section, we propose an automated EEG labeling method consisting of a first phase to detect the movement from EMG and a second phase to label EEG signals with an identified movement onset.

3.1. Movement detection from EMG

Fig. 2 shows an overview of the movement detection process using the EMG. The movements were characterized as localized bursts in the EMG signals (see Fig. 2, raw EMG). In this figure, the raw EMG signal drifts as the signal recording proceeds and contains considerable noise. Therefore, we first conditioned the raw EMG signals to improve the signal-to-noise ratio (SNR) and then detected movements using our proposed clustering algorithm.

3.1.1. EMG preprocessing

Following the existing work [18], we proposed three steps for EMG preprocessing. Specifically, we first applied a sixth-order Butterworth bandpass filter at 30–300 Hz to eliminate the baseline drift and movement artifacts in the raw EMG signals. Then, we applied the Teager–Kaiser energy operator (TKEO) rectification to enhance the SNR. TKEO [19,20] is the key to the preprocessing step. It considers both the amplitude and frequency of motor unit action potentials as follows [18]:

$$\hat{x}_t = x_t^2 - x_{t+1} \cdot x_{t-1}, \tag{1}$$

where x_t is the EMG amplitude at time t. Finally, we applied a second-order Butterworth low-pass filter at 50 Hz to reduce high-frequency noise.

3.1.2. Movement detection via clustering

We proposed a clustering approach for movement detection from preprocessed EMG (see Fig. 2, right). It uses a dynamic threshold to recognize the EMG signals of movement and then clusters nearby EMG movement signals to indicate complete movement instances. A movement instance is a period of consecutive signals of the movement during which a participant is executing a movement task, either opening or closing.

Recognize EMG signals of movement On the preprocessed EMG signals, we generated a dynamic threshold as follows:

$$thld = 1.2\mu + 2\sigma, \tag{2}$$

where μ is the mean and σ is the standard deviation of a trace of the EMG signals. We used this threshold to recognize the EMG signals that should be considered as movements. If a signal point was larger than the threshold, it was regarded as a signal of movement.

Cluster EMG signals of movement. After obtaining the movement signals, we clustered nearby signals to recognize complete movement instances. Specifically, we checked the time interval between two consecutive movement signal points. If the time interval is less than a distance parameter d, we added the signal point to the cluster currently being formed; otherwise, we considered the signal point as the first signal point of a new cluster. We estimated d based on the data collection process, which is approximately 5 s. Then, we gradually adjusted d until we detected approximately 20 clusters.

However, the detected clusters are not necessarily correct; therefore, we further apply a proximity heuristic to refine the detected clusters. We merged two close clusters into a single

cluster. The distance between the end of a cluster and the start of the next cluster was calculated for every cluster in the trace. This process merges any two clusters that are less than 50% of the median distance.

Outlier removal and normalization. Although we performed preprocessing to remove noise and artifacts, some outliers still remained. Therefore, we proposed removing outlier clusters based on their relative sizes (i.e., the number of movement signal points in the cluster). Specifically, we removed the clusters that were smaller than 10% of the median cluster size. The 10% value was empirically determined through testing.

Then, we normalized the clusters by reducing any cluster whose peak, which is the highest rectified signal point in terms of amplitude, was larger than the median value of the first quantile (Q1) of all the clusters' peaks. This was repeated until all the clusters' signal points are less than Q1. We performed normalization to reduce the amplitude discrepancy in the signal. This is particularly beneficial in the test traces, where opening and closing movements are present in the same trace. These movements present different amplitudes on the muscle from which we recorded the EMG signals.

Rerecognize and recluster signals of movement. As shown in Fig. 2, we used the normalized EMG signals to recognize the new threshold for movement detection and performed clustering again. However, we skipped the outlier removal and normalization in subsequent iterations. This process allows the clusters to encapsulate more of a movement, usually more of the start and end of the movement. Finally, different movement instances are identified from the EMG as shown in Fig. 2, where different colors indicate different movement instances in the top right schematic. Algorithm 1 shows the entire process of EMG signal clustering process for movement detection.

3.2. EEG labeling

After the EMG signals of movement were obtained, we labeled the EEG signals to create training samples with movement onset. We used the *movement onset*, which is the first data point in a movement instance (see Fig. 3), to denote a movement. The MRCP naturally appears along with movement initiation, making the movement onset the desirable annotation point. As shown in Fig. 3, we introduced a sliding window with a size of 2 s and a sliding step of 100 ms. This window size ensures that we can cover the expected parts of the MRCP at the movement onset. Whenever the sliding window intersects with a movement onset, we labeled the entire sliding window as a movement and considered it a movement sample. Thus, we created a total of 20 (i.e., sliding window size/step) movement samples per recognized movement instance.

4. Online evaluation

Most current EEG-based BCI studies rely on offline evaluations, which mainly test the accuracy of machine learning models or feature extraction approaches. However, this does not necessarily reflect the actual performance of a BCI system. On the other hand, traditional online evaluation depends on participants undergoing additional tests and requires more effort by the technicians. Additionally, studies based on public datasets may not have access to the same equipment used to collect the data; hence, they cannot evaluate their BCI in an online setting. Therefore, we proposed a pseudo-online evaluation suite that mimics the real-time procedure in which movement detection was performed using continuous data. Fig. 4 shows the schematic of our pseudo-online evaluation pipeline.

Algorithm 1 Movement Detection via Clustering

Input: a trace of preprocessed EMG signals $X = \{\hat{x}\}$ and associated time stamps $T = \{t\}$
Output: time stamps of EMG movement instances T_L
    # Recognize EMG points of movement
1: $thld = 1.2\mu(X) + 2\sigma(X)$ ▷ calculate the threshold
2: $X_m \leftarrow []$ ▷ initialize an empty list of signal points of movement
3: $T_m \leftarrow []$ ▷ initialize an empty list of associated time stamps
4: **for** $x, t \text{ in } X, T$ **do**
5:     **if** $x > thld$ **then**   ▷ find signal points of movement
6:         $X_m.insert(x), T_m.insert(t)$
7:     **end if**
8: **end for**   ▷ return X_m and T_m
    # Cluster EMG signals of movement
9: $L, T_L \leftarrow [], []$ ▷ initialize empty lists of movement instances
10: $C, T_C \leftarrow [X_m[0]], [T_m[0]]$ ▷ initialize a candidate movement instance
11: $i = 0$
12: **while** $i < len(X_m)$ **do**
13:     **if** $T_m[i+1] - T_m[i] < d$ **then**   ▷ form an instance by proximity
14:         $C.insert(X_m[i+1]), T_C.insert(T_m[i+1])$
15:     **else**   ▷ end the current movement instance
16:         $L.insert(C), T_L.insert(T_C)$
17:         $C, T_C \leftarrow [X_m[i+1]], [T_m[i+1]]$ ▷ start a new movement instance
18:     **end if**
19:     $i = i + 1$
20: **end while**   ▷ return L and T_L
    # Merge close movement instances
21: $median \leftarrow 0.5 \cdot \text{MEDIAN}(\{T_L[i+1]_{start} - T_L[i]_{end}\})$
22: i = 0
23: **while** $i < len(L)$ **do**
24:     **if** $T_L[i+1]_{start} - T_L[i]_{end} < median$ **then**
25:         $L[i].append(L[i+1]), T_L[i].append(T_L[i+1])$
26:     **end if**
27:     $i = i + 1$
28: **end while**   ▷ return L and T_L
    # Outlier removal and normalization
29: $size \leftarrow 0.1 \cdot \text{MEDIAN}(\{len(L[i])\})$
30: $i = 0$
31: **while** $i < len(L)$ **do**
32:     **if** $len(L[i]) < size$ **then**   ▷ outlier detection by size
33:         $L.delete(L[i]), T_L.delete(T_L[i])$
34:     **end if**
35:     $i = i + 1$
36: **end while**   ▷ return L and T_L
37: **while** $\forall L[i], L[i]_{peak} > Q1$ **do**
    ▷ Q1 = median value of first quartile of $\{L[i]_{peak}|C \in L\}$
38:     $L[i] \leftarrow 0.5 \cdot L[i]$ ▷ normalization by first quartile
39: **end while**   ▷ return L and T_L
40: repeat line 1 to line 28 with updated X from L

4.1. Evaluation pipeline

As shown in Fig. 4, the proposed pseudo-online evaluation pipeline has two components: data preparation and movement detection. In the data preparation component, compared to offline evaluations in which the entire EEG trace is acquired at once, the EEG data was incrementally generated for the online evaluation. Considering this, we implemented a data buffer to store a segment of the input EEG data and to process the buffered data. Specifically, we incrementally accumulated data in the buffer and updated it with increments at the tail every 100 ms as the buffer moved forward (see Fig. 4). For the preprocessing, we performed a second-order Butterworth band-pass filter at 0.5–4 Hz on the
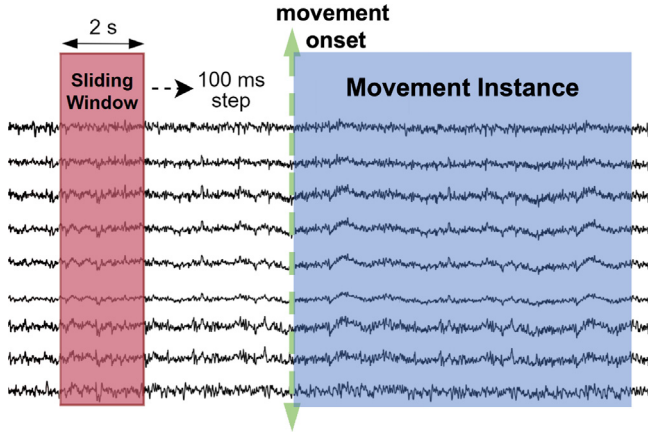
Fig. 3. EEG labeling with a sliding window. When the sliding window intersects with the movement onset, it is labeled as movement otherwise rest.

buffered data. This filter corresponds to the delta band, where the dominant frequency range of the MRCP lies. We also performed normalization on the filtered data as the last preprocessing step. When a buffer of preprocessed data is ready, we applied a sliding window of 2 s at the frontier of the data buffer for the subsequent movement detection.

If the preprocessing of the buffered data takes longer than the buffer update speed (i.e., 100 ms), the system cannot operate in real time. Thus, the data buffer length is a tradeoff between filtering the signal uniformly and processing the signal within the time constraint of buffer update speed. A data buffer that is considerably short may not capture enough data to represent the entire trace. In contrast, a data buffer that is considerably long would increase the computational complexity, resulting in a preprocessing time longer than the buffer update speed. Therefore, through empirical evaluation, we set the buffer size to be 20 s, which can sufficiently handle both sides of the tradeoff.

For the movement detection component, we set two prerequisites before performing detection. First, we detected whether exists a blink exists within the sliding window using the EOG signal [21]. Blinks are known artifacts that distort the EEG signals, making it challenging for the classifier to detect MRCPs [22]. When a blink is detected in the sliding window, the movement detection is stopped until the blink is no longer in the current sliding window. Second, we checked whether the system is in the freezing time, which is a period during which the system executes predefined commands after a movement is detected. We considered a BCI-based soft glove control system as an example. When a movement is detected, the soft glove is closed and remains closed for a period of time (i.e., the freezing time), during which the data buffer continues to be updated but the movement detection is suspended. After the freezing time, the system would resume detecting the next movement to open the glove.

After checking the two prerequisites, the data are fed into the movement detection algorithm. We proposed a *dwell heuristic* to determine this movement. In particular, we counted the number of detected movements in a sequence of detection attempts; if it exceeds a threshold, a movement is determined, as shown in Fig. 4. The design of the dwell heuristic has two considerations: (1) if every movement detection triggers a movement execution, the system would be too sensitive; (2) the dwell heuristic would reduce the false execution rate because the machine learning detector accuracy cannot be 100%. Our dwell heuristic considers a detection queue (DQ) of the 10 most recent detection attempts. The threshold is called a *dwell parameter*, which is a means of adjusting the tradeoff between sensitivity and specificity. A small

dwell parameter triggers movement execution more frequently, increasing the true detection rate and making the system susceptible to a higher false detection rate. However, a large dwell parameter may reduce the false detection rate but result in more missed movements.

4.1.1. Dwell parameter calibration

Setting a proper dwell parameter is nontrivial. We performed an automatic dwell parameter calibration prior to the actual online evaluation on two unseen recorded training traces from a specific participant. We used the median dwell parameter of the DQ for all detections within which there is expected to be a movement. Algorithm 2 presents the dwell-parameter calibration procedure. For each DQ, we have a corresponding label queue (LQ) that holds the ground truth for each detection. The inputs $\hat{Y}$ and Y correspond to the chronological instances of the DQ and LQ, respectively. If a detection interacts with a movement onset, it is denoted as 1; otherwise, it is 0. If all detections in an LQ are 1, we counted and summed the detections in the corresponding DQ (lines 3–4 in Algorithm 2). The detected movement median of the eligible DQ is used as the dwell parameter (line 8 in Algorithm 2).

Algorithm 2 Dwell Parameter Calibration

Input: $\hat{Y}, Y$
Output: d_{well}
1: $D_{well} \leftarrow []$
2: **for** DQ, LQ **in** $\hat{Y}, Y$ **do**
3:     **if** SUM(DQ) = 10 **then**   ▷ find movement onset
4:         $D_{well}.insert($SUM(DQ))
5:     **end if**
6: **end for**
7: SORT(D_{well})
8: $d_{well} \leftarrow$ MEDIAN(D_{well})   ▷ use the median as the dwell parameter

---

4.2. Online evaluation metrics

Most current online BCI studies aim to evaluate machine learning models and use the accuracy as the evaluation metric. However, in an online system, a timely response is critical for smooth operations. Therefore, it is desirable to include additional indicators for evaluating the timeliness of online BCI systems. Considering that detection is often either slightly before or after movement onset, we proposed to measure the mean detection latency (MDL) from movement detection to its nearest movement onset. It is suitable for online evaluation because movement detection is valid in real-world applications if it occurs within an acceptable, user-defined time frame of actual occurrence. The MDL is defined as:

$$MDL = \frac{1}{n} \sum_{i=1}^{n} \min_{y \in Y} |t_y - t_{\hat{y}_i}|, \tag{3}$$

where for each movement detection $\hat{y}_i$ from the model, we found the minimum distance in time to its nearest movement onset label y, and Y denotes the set of labels in the trace.

5. Experiments and results

In this section, we present experiments to evaluate our proposed automated labeling method on our collected dataset. Then, we explore popular machine/deep learning methods for self-paced movement detection under both offline and online settings.
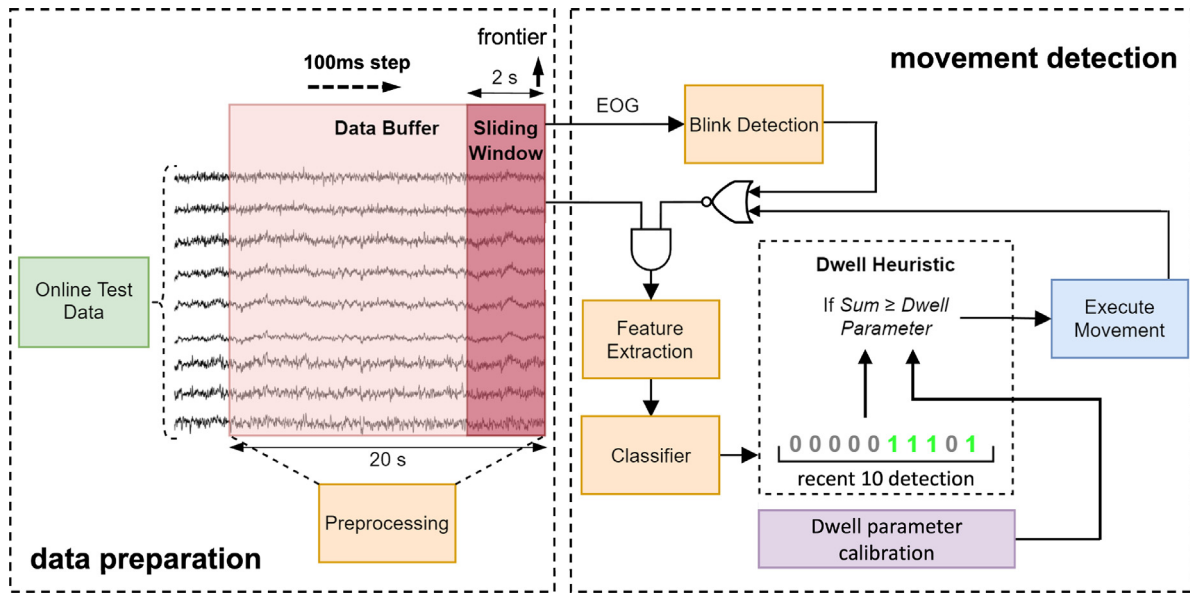
Fig. 4. The proposed pseudo-online evaluation pipeline. It has two components: data preparation and movement detection. (1) Data preparation: We build a data buffer to accumulate continuously incoming EEG data for uniform preprocessing. The front two seconds of the data buffer are considered as the sliding window for subsequent movement detection. (2) Movement detection: For each sliding window, we perform blink detection. If a blink occurs, we halt the detection until the sliding window has moved past the blink. We consider the ten most recent detection. If their sum exceeds the dwell parameter, the system executes a movement, and we halt the process for a period of freezing time.

5.1. Automated labeling evaluation

We built a baseline labeling method to demonstrate the superiority of our proposed automated labeling method. To perform the labeling, the baseline method only uses the first two intuitive steps–*recognizing EMG signals of movement* and *clustering EMG signals of movement*. Table 1 summarizes the results. The evaluation set had 440 movement instances per participant, including 400 (20 traces) for training and 40 (2 traces) for testing.

It is observed that our proposed labeling method can recognize all movement instances for all participants. In contrast, the baseline method could only recognize all movement instances for four out of ten participants (Sub 0, 2, 7, and 9). We used underlining to highlight the results that the baseline method fails to recognize all movement instances. It is also shown that the baseline method missed a maximum of 14 movements. In addition, not all recognized movement instances can reflect the actual movement. Fig. 5 shows the labeling results of both our proposed and baseline labeling methods for Sub 5 as an example. Colored bursts indicate recognized movement instances. It shows that our proposed labeling method successfully recognizes 20 movement instances, which is the desired result, whereas the baseline method can only recognize 19 movement instances where some clusters encompass multiple EMG bursts (see the bursts around 80–100 s in Fig. 5(b)). Furthermore, the last movement recognized by the baseline method is an erroneous recognition. Our proposed automated labeling method contends better with the noise in EMG signals owing to the extra steps and repeated clustering required to condition the signals, yielding better movement recognition and thus EEG labeling.

5.2. Self-paced movement detection

After data labeling, we explored self-paced movement detection with popular machine/deep learning methods under both offline and online settings.

Table 1
The number of movement instances labeled for each participant using our automated labeling method and the baseline method. The ground truth is 440 movements per participant.

	Our proposed	Baseline	Ground truth
Sub 0	440	440	440
Sub 1	440	426	440
Sub 2	440	440	440
Sub 3	440	434	440
Sub 4[a]	440	436	440
Sub 5	440	428	440
Sub 6	440	439	440
Sub 7	440	440	440
Sub 8	440	438	440
Sub 9	440	440	440

[a]Denotes a left-handed participant.

5.2.1. Experimental settings

We investigated well-documented and established EEG-BCI classifiers and popular machine/deep learning methods, namely (1) CSP-SVM [23,24], (2) DeepConvNet [25], (3) ShallowConvNet [25], (4) EEGNet [26], (5) ViTransformer (ViT) [27], and (6) XGBoost [28].

All deep learning models were implemented using their original code with mandatory modifications for our EEG data. They were trained for 300 epochs by default, but with possible early stopping. The CSP-SVM model was implemented using widely used tools.[2] Among all baselines, only XGBoost is a pure classifier that might be limited in feature extraction and, thus, the final detection. Therefore, we designed handcrafted features instead of raw signals as input into XGBoost to ensure competitive performance. We defined three statistical features as follows:

$$\bar{\hat{x}} = \frac{1}{n} \sum_{i=i}^{n} \hat{x}_i, \tag{4}$$


Table 2

Offline evaluation accuracy of 5-fold cross-validation. For a right-handed participant, the Primary Movement Hemisphere is the left hemisphere (C5, C3, C1), and for a left-handed participant, it is the right hemisphere (C2, C4, C6). We highlight the best results in bold.

5-fold CV	All channels										
Participants	Sub 0	Sub 1	Sub 2	Sub 3	Sub 4[a]	Sub 5	Sub 6	Sub 7	Sub 8	Sub 9	Avg
CSP-SVM	59.2 ± 0.8	53.6 ± 1.5	55.7 ± 0.7	55.7 ± 0.7	59.4 ± 0.4	56.6 ± 0.1	55.7 ± 0.4	59.5 ± 0.1	57.1 ± 0.3	54.3 ± 0.7	56.7 ± 0.6
DeepConv	61.1 ± 0.1	71.3 ± 0.3	74.2 ± 0.3	63.3 ± 0.3	66.7 ± 0.2	66.8 ± 0.2	56.3 ± 0.2	71.3 ± 0.1	74.4 ± 0.1	72.2 ± 0.1	67.8 ± 0.2
ShallowConv	58.8 ± 0.3	56.0 ± 0.1	59.1 ± 0.1	52.0 ± 0.1	60.3 ± 0.1	56.3 ± 0.2	55.1 ± 0.2	63.3 ± 0.1	61.2 ± 0.2	56.0 ± 0.3	57.8 ± 0.2
EEGNet	62.1 ± 0.9	62.1 ± 0.7	66.2 ± 0.9	63.3 ± 3.1	65.1 ± 0.7	64.1 ± 1.0	55.1 ± 2.9	68.8 ± 0.4	70.1 ± 0.5	69.4 ± 0.3	64.6 ± 1.2
ViT	64.5 ± 0.1	58.4 ± 0.7	66.1 ± 1.0	58.5 ± 0.2	67.9 ± 0.9	63.6 ± 0.6	65.4 ± 0.7	69.8 ± 0.7	74.8 ± 1.1	63.4 ± 0.7	65.2 ± 0.6
XGBoost	$\mathbf{86.9 \pm 0.1}$	$\mathbf{85.9 \pm 0.1}$	$\mathbf{88.1 \pm 0.8}$	$\mathbf{86.3 \pm 0.4}$	$\mathbf{88.8 \pm 0.5}$	$\mathbf{84.1 \pm 0.6}$	$\mathbf{86.5 \pm 0.1}$	$\mathbf{87.1 \pm 0.1}$	$\mathbf{88.9 \pm 0.6}$	$\mathbf{86.4 \pm 0.9}$	$\mathbf{86.9 \pm 0.4}$

5-fold CV	Primary movement hemisphere										
CSP-SVM	58.8 ± 0.8	51.9 ± 0.1	55.1 ± 0.4	52.3 ± 0.6	58.1 ± 0.3	57.1 ± 0.2	55.6 ± 1.0	59.2 ± 0.8	57.1 ± 0.5	53.2 ± 0.9	55.8 ± 0.6
DeepConv	59.7 ± 1.0	56.1 ± 0.8	64.1 ± 0.1	54.8 ± 2.0	61.8 ± 2.0	59.0 ± 6.4	55.0 ± 1.0	62.7 ± 1.0	64.9 ± 0.1	63.4 ± 0.7	60.2 ± 1.2
ShallowConv	53.3 ± 1.2	50.9 ± 0.8	55.9 ± 2.0	49.0 ± 1.1	55.1 ± 1.0	54.2 ± 4.1	52.1 ± 2.0	56.3 ± 2.1	61.6 ± 2.5	51.9 ± 1.2	54.0 ± 1.8
EEGNet	58.9 ± 0.1	61.3 ± 0.9	64.2 ± 0.7	58.4 ± 0.9	61.6 ± 0.5	64.1 ± 0.7	54.1 ± 1.8	62.2 ± 1.7	70.6 ± 1.4	63.6 ± 1.4	61.9 ± 1.0
ViT	62.9 ± 1.1	53.1 ± 1.5	58.9 ± 0.6	52.1 ± 0.7	62.2 ± 0.8	57.7 ± 0.3	58.1 ± 0.6	67.7 ± 0.4	66.1 ± 1.3	59.4 ± 1.3	59.8 ± 0.9
XGBoost	$\mathbf{82.2 \pm 0.4}$	$\mathbf{78.3 \pm 2.1}$	$\mathbf{83.3 \pm 0.3}$	$\mathbf{80.7 \pm 0.5}$	$\mathbf{84.3 \pm 0.6}$	$\mathbf{76.7 \pm 1.5}$	$\mathbf{80.3 \pm 1.3}$	$\mathbf{83.8 \pm 0.9}$	$\mathbf{84.6 \pm 0.7}$	$\mathbf{80.1 \pm 1.3}$	$\mathbf{81.4 \pm 1.0}$

[a]Denotes a left-handed participant.
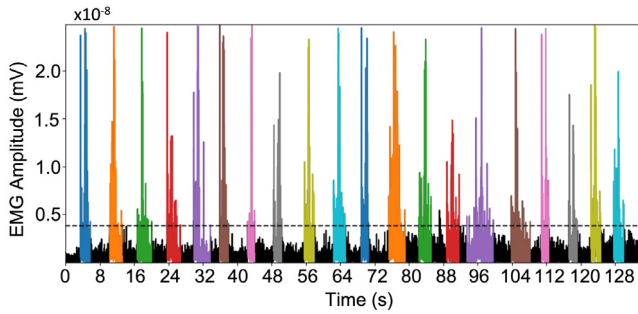
(a) Our proposed automated labeling method.
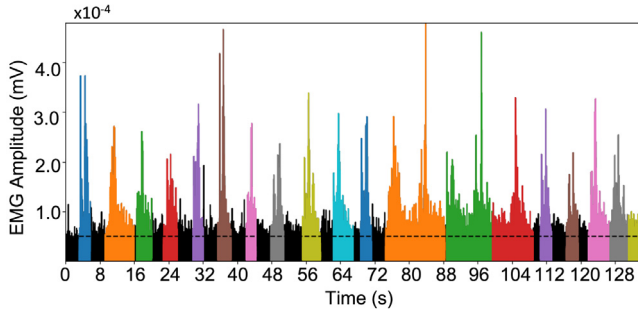
(b) Baseline labeling method.

Fig. 5. Comparison between (a) our proposed automated labeling method, and (b) the baseline labeling method. Both methods were applied to the same trace from Sub 5.

$$\hat{x}_{kurtosis} = \frac{1}{n} \frac{\sum_{i=1}^{n}(\hat{x}_i - \bar{\hat{x}})^4}{\sigma^4}, \tag{5}$$

$$\hat{x}_{skewness} = \frac{1}{n} \frac{\sum_{i=1}^{n}(\hat{x}_i - \bar{\hat{x}})^3}{\sigma^3}, \tag{6}$$

where n is the number of data points in the sliding window. We defined two temporal features as:

$$t_{slope} = t_{max} - t_{min}, \tag{7}$$

where t_{max} and t_{min} are the time points within the sliding window with the highest and lowest amplitude values, respectively. The other temporal feature is defined as a slope given by:

$$\hat{x}_{slope} = \frac{\hat{x}_{max} - \hat{x}_{min}}{t_{slope}}. \tag{8}$$

Table 3

The t-test results of comparing using all channels and PMH channels for offline evaluation. We highlight significant *p*-values ($p < 0.05$) in bold. D-Conv and S-Conv represent DeepConv and ShallowConv, respectively.

	CSP-SVM	D-Conv	S-Conv
p-value	0.222	**0.001**	**0.011**
	EEGNet	ViT	XGBoost
p-value	0.127	**0.012**	<0.001

5.2.2. Offline evaluation

In the offline evaluation, we explored two channel settings: (1) using all available channels and (2) using only the channels placed over the primary movement hemisphere (PMH). For a right-handed participant, the PMH was in the left hemisphere (C5, C3, C1), and for a left-handed participant, it was in the right hemisphere (C2, C4, C6) [29,30]. We illustrated the results using five-fold cross-validation on the training data for each participant presented in Table 2. Cross-validation is based on signal traces. In each validation, 16 traces were used for training and four traces were used for validation.

The accuracy of XGBoost using handcrafted features consistently outperforms all other methods on all participants. Moreover, it is also observed that the performance in terms of both accuracy and standard deviation on PMH is consistently worse than when using all channels. We conducted a statistical t-test to validate the significance of the comparison between using all channels and PMH. The *p*-value results are presented in Table 3, with significant results ($p < 0.05$) shown in bold. Four out of the six approaches indicate that using all channels is significantly better than using the PMH only. We hypothesized that the models benefit from the increased information across the movement cortex from all channels to obtain a slight advantage over using PMH only. We further investigated this finding in an online evaluation.

5.2.3. Online evaluation

We performed an online evaluation of movement detection using our proposed pseudo-online evaluation pipeline. Table 4 shows the online evaluation results on the two test traces described in Section 2. *Movement*(%) estimates the completeness of a BCI system, which is the percentage of correct detection out of the total actual movements. *Precision*(%) measures the percentage of correct detection out of the total number of detections being made. It is shown that XGBoost outperforms other machine learning models in terms of *Movement*(%) and *Precision*(%), which is the same as offline evaluation. However, the superiority margin

Table 4
Online evaluation results that are averaged over participants.

Channels Metric	All Movement (%)	PMH Movement (%)	All Precision (%)	PMH Precision (%)	All MDL (s)	PMH MDL (s)	All Dwell	PMH Dwell
CSP-SVM	58.2 ± 16.5	56.1 ± 9.12	34.6 ± 11.5	36.1 ± 12.1	1.04 ± 0.25	0.95 ± 0.21	4.6 ± 3.5	6.7 ± 2.9
DeepConv	53.0 ± 10.4	51.5 ± 12.1	**40.5 ± 11.0**	39.6 ± 10.7	**0.77 ± 0.25**	**0.82 ± 0.25**	5.8 ± 2.6	7.1 ± 2.0
ShallowConv	53.1 ± 11.1	52.7 ± 17.8	38.7 ± 11.3	35.7 ± 11.3	0.91 ± 0.28	0.89 ± 0.27	5.8 ± 3.1	4.4 ± 2.9
EEGNet	48.6 ± 10.7	51.7 ± 17.6	40.2 ± 8.66	38.4 ± 10.3	0.78 ± 0.21	0.88 ± 0.19	8.2 ± 1.9	6.7 ± 3.2
ViT	51.5 ± 8.74	50.0 ± 7.15	36.6 ± 10.2	37.3 ± 13.5	0.95 ± 0.22	0.83 ± 0.24	6.3 ± 1.3	6.8 ± 1.6
XGBoost	**60.6 ± 8.52**	**59.6 ± 9.23**	38.0 ± 12.1	**41.0 ± 14.1**	0.91 ± 0.15	0.94 ± 0.20	6.3 ± 0.9	6.3 ± 1.2

Table 5
The t-test results (p-value) of comparing using all channels and PMH channels for online evaluation. The significance level is $p < 0.05$.

	Movement	Precision	MDL
CSP-SVM	0.367	0.398	0.296
DeepConv	0.386	0.428	0.326
ShallowConv	0.475	0.287	0.433
EEGNet	0.329	0.348	0.091
ViT	0.343	0.450	0.131
XGBoost	0.119	0.316	0.367

is much smaller than that of the offline evaluation, and XGBoost is inferior to DeepConv for *Precision*(%) of *All* channels. Regarding detection timeliness, DeepConv has the least detection latency (MDL) and is much faster than XGBoost for both *All* channels and *PMH* channels. In addition, using *PMH* channels is worse than using *All* channels regarding *Movement*(%) and similar to using *All* channels regarding *Precision*(%), while the performance gaps are quite limited. We further presented the t-test comparison results of comparing using *All* channels versus only using *PMH* channels for the online evaluation results in Table 5; compared to the offline evaluation, no statistical significance was observed for any models in the online evaluation. Based on the above observations and considering that precision is much more important than the detection completeness of a BCI system, our online evaluation indicates that the DeepConv-based BCI system is the most suitable for online usage.

In addition to the observations above, we noticed that on an average the difference in the dwell parameter between these approaches was minor but not uniform across the same classifier, indicating the need to tune a dwell-specific parameter for each participant and the machine learning model. The proposed dwell parameter calibration approach can facilitate this parameter specification process.

5.3. Visualization

We visualized the online results of a test trace in Fig. 6 to better illustrate the online evaluation process and results. The green rectangles indicate correct detection, red rectangles denote FP detection, and dashed lines denote the movement onset. We showed the detected blinks as small squares above the predictions. As shown in Section 4, we suspended the detection when blinks were detected.

It is shown that generally the FP detection is relatively late to their nearest movement onset or inserted between two correct detections. Late detection is acceptable as long as its in a reasonable range, while inserted false detection should be avoided as it may incur severe consequences. In addition, there is some missing detection (see movements around 58 s). Missing detection would reduce the efficiency of the system, but not lead to potentially dangerous, erroneous executions. It is also noticed that there are many blinks that degrade the signal quality and thus require detection suspension.

6. Discussion

In this section, we discuss our main research findings and design considerations.

6.1. Self-paced data collection

In this study, we focused on self-paced BCI systems, which have the following advantages in contrast to conventional cue-based BCI systems: (1) self-paced BCI relies on naturally occurring cortical response (i.e., MRCP) and does not require training the users; (2) MRCP similarly occurs for both healthy and disabled people and for both movement imagination and execution; thus, the data can be collected from healthy people performing movement execution; (3) no external cues (e.g., visual cues) are required for labeling because the data can be collected from movement execution, and thus the obtained EEG signals are free from the noise incurred by the cues.

Participants perform a movement in a self-paced way, therefore the time interval between adjacent movements is inconsistent but around 5 s according to our instruction, as shown in Fig. 6. We did not strictly require subjects to rest for 5 s. This suggested that the rest time keeps the whole experiment time within a reasonable range, so as not to make the participants considerably tired. Although such a data collection process minimizes external interference and mimics real application scenarios, it poses the challenge of labeling the movement onset.

6.2. Automated labeling

To resolve the labeling challenge, we proposed an automated labeling approach to relieve the human burden and obtain better labeling quality. As shown in Fig. 5, although it is possible for human experts to roughly recognize movement range, it is difficult to precisely identify movement onset. Thus, a movement-onset detection algorithm is preferred. Moreover, there is considerable noise in EMG signals owing to line interference, unintended muscle movement, and environmental impedance. The naive baseline method was easily affected by noise. For example, the second orange cluster in Fig. 5(b) actually has two movement bursts, which is recognized as a single movement because of the noise between the two bursts. The same pattern was observed for the second green cluster in Fig. 5(b). In Fig. 5(a), it is clear that the rest periods (black areas) between approximately 72 s to 104 s have larger magnitudes than the other rest periods, which are the wrongly recognized periods by the baseline method, as shown in Fig. 5(b). This indicates the failure reason of the baseline method, while our method can overcome it by enhancing the SNR.

6.3. Online evaluation pipeline

Previous studies only use offline evaluation because of considerable extra efforts for online evaluation and the lack of equipment. Our online evaluation pipeline allows the estimation of a BCI's online performance without extra overhead. This is a
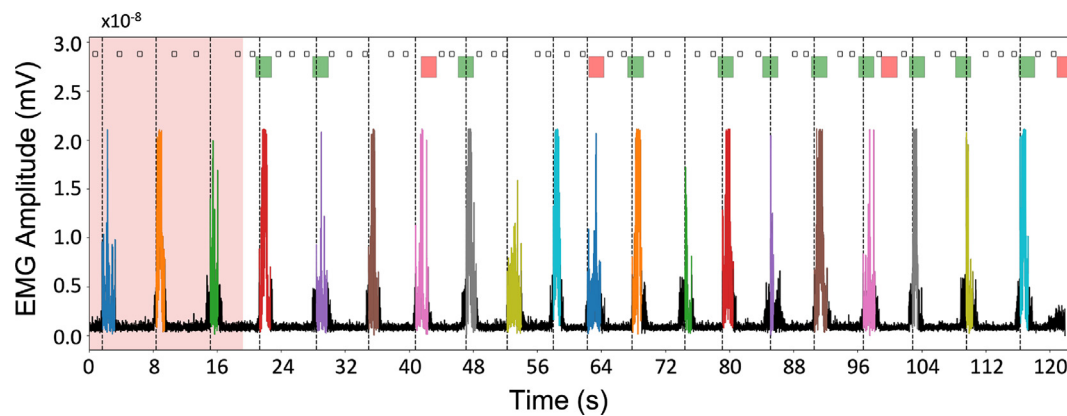
Fig. 6. Visualization of online evaluation on a test trace from Sub 7. Movement instances of EMG are shown in different colors and dashed lines denote the movement onset. The opaque red field shows the data buffer, which we build initially. The red and green rectangles denote the FP and correct detection. Their sizes correspond to their respective sliding windows. Detected blinks are shown as small squares above the detection.

step forward with respect to addressing the challenges of online evaluations in BCI studies.

In addition, we show that the offline results do not necessarily map to online results. According to Tables 2 and 4, the following was demonstrated that (1) the performance gap between the best model and the others of online evaluation is smaller than that of offline evaluation, and (2) the best model in offline evaluation is not the most suitable for online use, especially considering online detection latency. The offline and online results discrepancy is mainly owing to the *dwell heuristic* procedure. However, using the dwell heuristic is necessary because working in an online or real-life environment requires the reduction of FP detection, as they may cause detrimental misoperations. Our automatic heuristic parameter calibration differs from that proposed by Savić et al. [16], where the dwell parameter is empirically determined. Instead, our algorithmic implementation leverages historical data and statistical observations, which is more consistent and robust and free of expertise.

7. Conclusion

This study targets two challenges of self-paced BCI systems: EEG labeling during data collection and online evaluation. We proposed an automated labeling approach to detect EMG bursts and onsets to accurately cross-label EEG signals. Then, we designed a pseudo-online evaluation pipeline to mimic real-world BCI systems and proposed a dwell heuristic to reduce false positive detection with an automatically determined dwell parameter. We collected data from 10 healthy subjects for the experiments. Our automated labeling results can outperform the naive labeling method and successfully detect all EMG bursts. Moreover, the offline results do not necessarily map to online results because of latency consideration and the dwell heuristic of reducing false positive detection. This study contributes to the community of self-paced BCI research by alleviating labor requirements in both data collection and online evaluation.

CRediT authorship contribution statement

Dalin Zhang: Conceptualization, Methodology, Writing, Resources, Supervision. **Christoffer Hansen:** Software, Validation, Methodology, Writing. **Fredrik De Frène:** Software, Validation, Methodology, Writing. **Simon Park Kærgaard:** Software, Validation, Methodology, Writing. **Weizhu Qian:** Writing – review & editing. **Kaixuan Chen:** Conceptualization, Writing – review & editing, Resources.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] R. Abiri, S. Borhani, E.W. Sellers, Y. Jiang, X. Zhao, A comprehensive review of EEG-based brain–computer interface paradigms, J. Neural Eng. 16 (1) (2019) 011001.

[2] D. Priyasad, T. Fernando, S. Denman, S. Sridharan, C. Fookes, Affect recognition from scalp-EEG using channel-wise encoder networks coupled with geometric deep learning and multi-channel feature fusion, Knowl.-Based Syst. (2022) 109038.

[3] D. Zhang, L. Yao, K. Chen, S. Wang, Ready for use: subject-independent movement intention recognition via a convolutional attention model, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1763–1766.

[4] A. Cruz, G. Pires, A. Lopes, C. Carona, U.J. Nunes, A self-paced BCI with a collaborative controller for highly reliable wheelchair driving: Experimental tests with physically disabled individuals, IEEE Trans. Hum.-Mach. Syst. 51 (2) (2021) 109–119.

[5] K.K. Ang, C. Guan, EEG-based strategies to detect motor imagery for control and rehabilitation, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (4) (2016) 392–401.

[6] B.A. Akinnuwesi, K.A. Olayanju, B.S. Aribisala, S.G. Fashoto, E. Mbunge, M. Okpeku, P. Owate, Application of support vector machine algorithm for early differential diagnosis of prostate cancer, Data Sci. Manag. (2022).

[7] G. Fan, Z. Deng, Q. Ye, B. Wang, Machine learning-based prediction models for patients no-show in online outpatient appointments, Data Sci. Manag. 2 (2021) 45–52.

[8] D. Zhang, L. Yao, K. Chen, S. Wang, P.D. Haghighi, C. Sullivan, A graph-based hierarchical attention model for movement intention detection from EEG signals, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (11) (2019) 2247–2253.

[9] H. Altaheri, G. Muhammad, M. Alsulaiman, S.U. Amin, G.A. Altuwaijri, W. Abdul, M.A. Bencherif, M. Faisal, Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: a review, Neural Comput. Appl. (2021) 1–42.

[10] D. Zhang, L. Yao, S. Wang, K. Chen, Z. Yang, B. Benatallah, Fuzzy integral optimization with deep q-network for eeg-based intention recognition, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 156–168.

[11] A. Al-Saegh, S.A. Dawwd, J.M. Abdul-Jabbar, Deep learning for motor imagery EEG-based classification: A review, Biomed. Signal Process. Control 63 (2021) 102172.

[12] H. Raza, D. Rathee, S.-M. Zhou, H. Cecotti, G. Prasad, Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface, Neurocomputing 343 (2019) 154–166.

[13] D. Zhang, K. Chen, D. Jian, L. Yao, S. Wang, P. Li, Learning attentional temporal cues of brainwaves with spatial embedding for motion intent detection, in: 2019 IEEE International Conference on Data Mining, ICDM, IEEE, 2019, pp. 1450–1455.

[14] M. Jochumsen, M.S. Navid, R.W. Nedergaard, N. Signal, U. Rashid, A. Hassan, H. Haavik, D. Taylor, I.K. Niazi, Self-paced online vs. Cue-based offline brain-computer interfaces for inducing neural plasticity, Brain Sci. 9 (6) (2019) 127.

[15] S. Aliakbaryhosseinabadi, S. Dosen, A.M. Savic, J. Blicher, D. Farina, N. Mrachacz-Kersting, Participant-specific classifier tuning increases the performance of hand movement detection from EEG in patients with amyotrophic lateral sclerosis, J. Neural Eng. 18 (5) (2021) 056023.

[16] A.M. Savić, S. Aliakbaryhosseinabadi, J.U. Blicher, D. Farina, N. Mrachacz-Kersting, S. Došen, Online control of an assistive active glove by slow cortical signals in patients with amyotrophic lateral sclerosis, J. Neural Eng. 18 (4) (2021) 046085.

[17] A. Valenti, M. Barsotti, D. Bacciu, L. Ascari, A deep classifier for upper-limbs motor anticipation tasks in an online BCI setting, Bioengineering 8 (2) (2021).

[18] S. Solnik, P. Rider, K. Steinweg, P. DeVita, T. Hortobágyi, Teager–Kaiser energy operator signal conditioning improves EMG onset detection, Eur. J. Appl. Physiol. 110 (3) (2010) 489–498.

[19] J.F. Kaiser, On a simple algorithm to calculate the 'energy' of a signal, in: International Conference on Acoustics, Speech, and Signal Processing, IEEE, 1990, pp. 381–384.

[20] J.F. Kaiser, Some useful properties of teager's energy operators, in: 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, IEEE, 1993, pp. 149–152.

[21] M. Agarwal, R. Sivakumar, Blink: A fully automated unsupervised algorithm for eye-blink detection in eeg signals, in: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2019, pp. 1113–1121.

[22] W.-D. Chang, H.-S. Cha, K. Kim, C.-H. Im, Detection of eye blink artifacts from single prefrontal channel electroencephalogram, Comput. Methods Programs Biomed. 124 (2016) 19–30.

[23] Z.J. Koles, M.S. Lazar, S.Z. Zhou, Spatial patterns underlying population differences in the background EEG, Brain Topogr. 2 (4) (1990) 275–284.

[24] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[25] R.T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, T. Ball, Deep learning with convolutional neural networks for EEG decoding and visualization, Human Brain Mapp. 38 (11) (2017) 5391–5420.

[26] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces, J. Neural Eng. 15 (5) (2018) 056013.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, ICLR, 2021, pp. 1–12.

[28] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[29] T. Alotaiby, F.E.A. El-Samie, S.A. Alshebeili, I. Ahmad, A review of channel selection algorithms for EEG signal processing, EURASIP J. Adv. Signal Process. 2015 (1) (2015) 66.

[30] M. Rashid, N. Sulaiman, A. PP Abdul Majeed, R.M. Musa, B.S. Bari, S. Khatun, et al., Current status, challenges, and possible solutions of EEG-based brain-computer interface: a comprehensive review, Front. Neurorobot. (2020) 25.