Biomedical Signal Processing and Control 86 (2023) 105202

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Ensemble deep clustering analysis for time window determination of event-related potentials

Reza Mahini [a], Fan Li [b], Mahdi Zarei [f], Asoke K. Nandi [c], Timo Hämäläinen [a,*], Fengyu Cong [a,b,d,e,*]

[a] Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
[b] School of Biomedical Engineering, Faculty of Electronic and Electrical Engineering, Dalian University of Technology, China
[c] Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge UB8 3PH, UK
[d] School of Artificial Intelligence, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China
[e] Key Laboratory of Integrated Circuit and Biomedical Electronic System, Liaoning Province. Dalian University of Technology, Dalian, China
[f] Department of Bioengineering and Therapeutic Sciences, and Programs in Biological Sciences and Human Genetics, University of California, San Francisco, CA 94158, USA

ARTICLE INFO

Keywords:
Event-related potentials
Time window
Deep clustering
Ensemble learning
Consensus clustering
ERP microstates

ABSTRACT

Objective: Cluster analysis of spatio-temporal event-related potential (ERP) data is a promising tool for exploring the measurement time window of ERPs. However, even after preprocessing, the remaining noise can result in uncertain cluster maps followed by unreliable time windows while clustering via conventional clustering methods.

Methods: We designed an ensemble deep clustering pipeline to determine a reliable time window for the ERP of interest from temporal concatenated grand average ERP data. The proposed pipeline includes semi-supervised deep clustering methods initialized by consensus clustering and unsupervised deep clustering methods with end-to-end architectures. Ensemble clustering from those deep clusterings was used by the designed adaptive time window determination to estimate the time window.

Results: After applying simulated and real ERP data, our method successfully obtained the time window for identifying the P3 components (as the interest of both ERP studies) while additional noise (e.g., adding 20 dB to −5 dB white Gaussian noise) was added to the prepared data.

Conclusion: Compared to the state-of-the-art clustering methods, a superior clustering performance was yielded from both ERP data. Furthermore, more stable and precise time windows were elicited as the noise increased.

Significance: Our study provides a complementary understanding of identifying the cognitive process using deep clustering analysis to the existing studies. Our finding suggests that deep clustering can be used to identify the ERP of interest when the data is imperfect after preprocessing.

1. Introduction

Event-related potentials (ERPs) data is a rich source of information about the cognitive process in the human brain. Information processing units are known as ERP components (i.e., particularly emerge as the ERP peaks). Qualifying ERP of interest for measuring the cognitive process is the key element for reporting results of processing ERP data and testing the research hypothesis. The conventional method for identifying an ERP is to measure the ERP's peak latency or the latency's mean amplitude in the time window measurement interval [30]. The conventional method for selecting the time window is primarily performed via visual inspection for the prominent peak amplitude or obtaining significant differences between the conditions/groups [20,21]. Another popular method is moving time intervals commonly used in different resolutions to find a large effect size [41,50,64]. This method, however, can report the effect size obtained from high-frequency noise as a biased result. The problem with such measurements is that if the underlying assumption for selecting the time window is invalid, analyzing the peak latency, i.e., aiming to detect a larger effect size, can be misleading or result in a problematic estimation of the brain response.

Regardless of the experiment design, various uncertainties can be investigated while processing ERP data. First, there is no available

* Corresponding authors.
E-mail addresses: timo.t.hamalainen@jyu.fi (T. Hämäläinen), cong@dlut.edu.cn (F. Cong).

https://doi.org/10.1016/j.bspc.2023.105202
Received 28 October 2022; Received in revised form 5 May 2023; Accepted 21 June 2023
Available online 2 July 2023

standard aggregated parameter setting for preprocessing methods. For instance, artifact rejection via visual inspection [10] or artifact rejection employing independent component analysis (ICA) [34] and electroencephalography (EEG) referencing methods [66] are commonly uncertain. However, some advanced preprocessing methods implemented in popular software, e.g., FieldTrip [45] and systematic artifact removal methods [16], can somewhat improve data quality. Next, data inconsistency can occur by EEG recording conditions and devices (if more than one), the participants' cortical differences, and the response delay inconsistency in the trials and individual subjects. This can also be associated with the involved number of trials [7]. However, the ERP process (averaging the trials) by itself can somewhat reduce the issue of incoherency between the trials. Finally, new recording technology and devices may require new and more robust analysis method designs.

In recent decades, cluster analysis has emerged as a technical solution used in different aspects of EEG/ERP studies. For example, using fuzzy clustering for class discrimination of evoked potential (EP) waveforms [35] and clustering of principal component analysis (PCA) results [14] for decreasing the effect of noise in the ERP waveforms were introduced. The more conservative methods for ERP identification, reviewed by Kallionpää et al. [18], showed that the cluster-based nonparametric testing method (in the popular FieldTrip software) relies on the temporally adjacent time point (as a cluster) if a significant effect size is identified. Furthermore, spatial clustering, initially introduced by Pascual-Marqui [47], revealed that the brain state called a 'microstate' can be explained as a quasi-stable spatial configuration [27]. Hence, the topographical configuration for a neural response does not change in milliseconds, e.g., 80–120 ms [68].

Two popular cluster analyzing approaches were used for processing spatio-temporal ERP data. First, microstate analysis that assigns the ERP microstates, i.e., represented by global field power (GFP) or GFP maxima, to the template map obtained from clustering the grand average ERP [23,42,62]. The template maps are cluster maps with a high explained variance (e.g., 70% of variance). The post-hoc processing is required, such as smoothing and refining processes based on spatial correlation evaluation if the data is noisy [38]. This method, however, ignores the polarity of time points when assigning clusters. Despite using GFP and the winner-takes-all strategy in determining template maps in the microstates analysis, as argued in some research [11,55], the second group takes whole time points and polarity into account for clustering of spatio-temporal ERP. Recently, we discussed qualifying ERP of interest using consensus clustering as a reliable method for ERP data in different resolutions [31,32,33]. However, cluster analysis of noisy data can result in many noisy clusters and loss of the main components due to being sensitive to the data quality if inappropriate clustering is applied.

Considering the uncertainty of the data, deep learning powered by deep neural networks (DNNs) achieved tremendous success using multiple hidden layers, particularly for EEG data with different designs [3,9,56,67] and our previous work for sleep staging [28]. Roy et al. [52] reviewed a wide range of DNNs used to analyze EEG data. Deep clustering, by definition, is introduced as a method encouraging DNNs to learn cluster-oriented feature representation and clustering. Therefore, DNNs with an embedded clustering module are used with the aim of transforming data points into cluster-friendly representations [2,51]. Yet, two popular strategies have been introduced for deep clustering [2,39]. First, a two-step process in which the DNN is trained to learn initialized labels investigating non-clustering loss (i.e., only the DNN's loss is considered). Then, a clustering method (e.g., *k*-means) is applied to the transformed data in the latent space (i.e., cluster-friendly representation). Another approach uses a jointly training DNN and clustering to optimize clustering and the DNN's weights simultaneously, in which the deep clustering improves the labeling obtained from the clustering layer/module. Deep clustering for brain imaging has been used in some recent studies [49,56]. However, there is little discussion about unsupervised identifying ERP components in the literature.

In this study, we investigate the determinants of the time window of the ERP of interest from spatio-temporal ERP data with various additional noises. We design an ensemble clustering pipeline from two groups of deep clustering methods. Semi-supervised deep clustering methods have been used in which DNN models are trained to learn the labeling that is calculated by state-of-the-art consensus clustering. Unsupervised deep clustering methods are designed via end-to-end DNN architectures for learning the input signal (i.e., with a joint clustering, depending on design). A newly updated time window determination method has been used to qualify ERPs of interest from ensemble clustering results. On the other hand, DNNs are powerful tools for learning nonlinear properties of neuroimaging data and are tolerant to noise and fault [13,36]. This motivates us to apply DNN to learn the most efficient features compared to conventional handcrafted features (with extensive domain expertise). We applied our method to two different ERP data for qualifying P3 components. We demonstrate that the proposed pipeline reliably estimates the time window of the ERP of interest when there is noise in the data after preprocessing.

2. Materials and methods

This section describes the ERP datasets used for testing our method, the proposed method in detail, and the assessment performance metrics.

2.1. ERP data

In order to assess the proposed method, we employed two ERP data, simulated and real data. For the simulated data [31], we test the proposed method against our prior knowledge, i.e., about the spatial and temporal properties of pre-defined ERP components when more noise is added to the data. Likewise, for the real ERP data, we test our method for qualifying the ERP of interest in the prior study [19] when the existing noise in the data increases.

2.1.1. Simulated ERP data

The simulated data was conducted using the 'DipoleSimulator 3.3.0.2′ software from BESA Research (https://www.besa.de). To this end, first, we defined dipoles to generate six pre-defined components (i.e., P1, N1, P2, N2, P3, and N4) and the corresponding data from each component with a simulated scalp containing 65 electrodes for 20 subjects (one group). Once data was generated, the sampling rate of generated ERP data was 214 Hz and epoched from 100 ms pre-stimulus onset to 600 ms post-stimulus via the software (i.e., the dataset size is 65×150). The signal was interpolated to 429 Hz (i.e., dataset size is 65×300) to increase the resolution of the data to provide potentially more isolation accuracy of ERP components. This was done via electrode-wise increasing the original sample rate to a higher rate by inserting zeros into the signal and applying a finite impulse response (FIR) digital interpolation low-pass filter [44,58] to expand the signal. We showed the simulated ERP components' properties (spatial and temporal) and the combined waveform in Fig. 1 for reference.

In order to provide individual ERP data of the subjects, a random resampling interpolation method was applied by increasing the duration of the component with a maximum of 11.5 ms (5 time points $\times$ 2.3 ms), resulting in a new signal (for each dipole). Then a further random shift was performed for each dataset within $\pm$ 4.6 ms. Finally, a combined ERP dataset from two conditions (i.e., 'Cond1′ and 'Cond2′) was prepared using MATLAB code. Noteworthy that an additional strength (subjectively) is applied to some of the components' waveforms (e.g., N2 and P3) to provide a significant difference between conditions. The P3 component in this data refers to the positive response from 266 to 357 ms post-stimulus. Statistical amplitude power differences were measured at CPz/Cz electrode sites.

2.1.2. Real ERP data

The proposed method was applied to the real ERP data, i.e., the active visual oddball task study published by Kappenman et al. [19], to

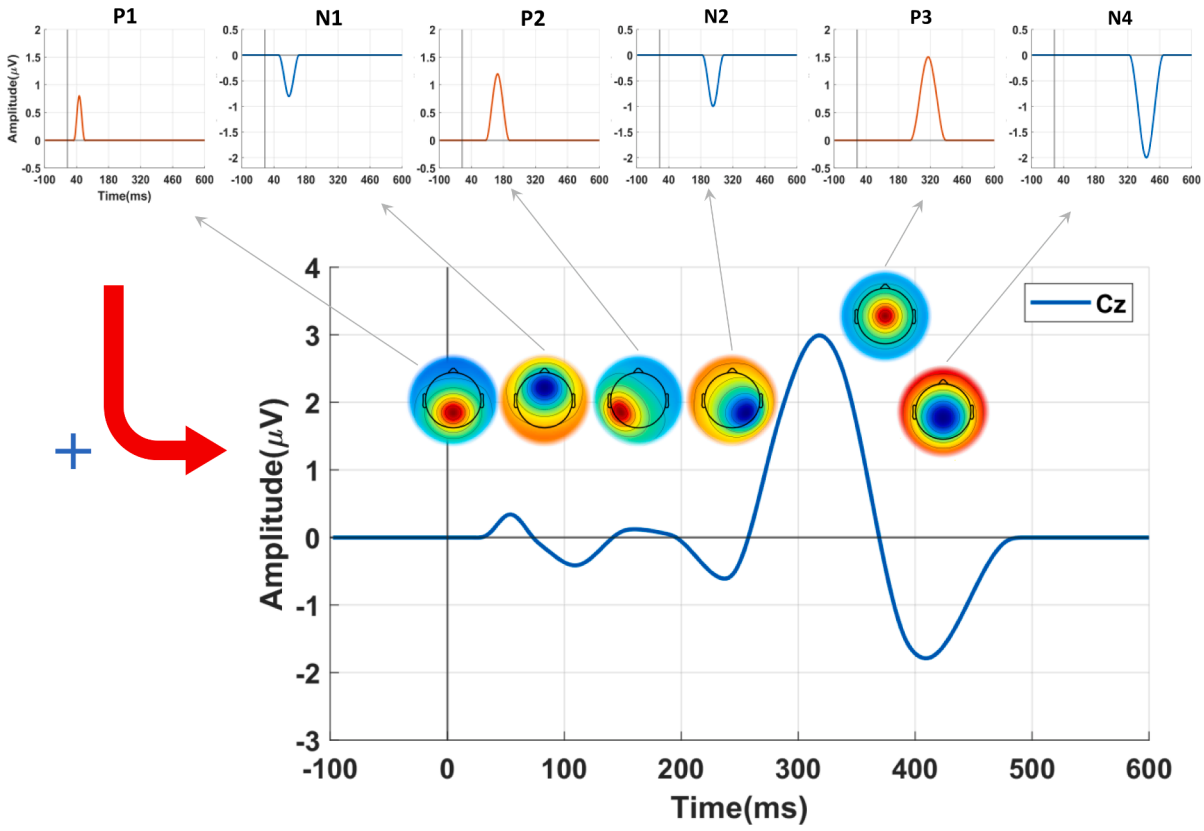

Fig. 1. Illustration of the simulated ERP components, corresponding topographical maps, and the combined waveform (in Cz electrode). The corresponding topographic maps of the pre-defined components are shown with the ERP waveform.

qualify the P3 component. The P3 component refers to the maximum positive peak around 300 to 600 ms (i.e., the rough time window for P3 from the prior study). The EEG data was recorded from 40 participants (i.e., 25 females and 15 males with a mean year of age = 21.5) with 30 scalp electrodes in the international 10/20 system from two conditions, 'Rare' and 'Frequent' letters. The recorded signals were digitized at 1024 Hz (sampling rate), downsampled to 256 Hz for faster processing, and referenced offline the average of P9 and P10. The location electrodes were excluded from processing (i.e., only 28 electrodes were considered). The signals were high-pass and low-pass filtered at 0.1 Hz and 20 Hz and epoched from 200 ms pre-stimulus onset to 800 ms post-stimulus onset by the experimenters. The experimenters extracted approximately 50 to 70 trials for each condition from each subject. The electrode Pz (state-of-the-art electrode) was considered for statistical power analysis following the prior research.

2.2. Our proposed method

Fig. 2 illustrates the proposed method in four steps, data preparation, consensus clustering, ensemble deep clustering, and time window determination. A more detailed explanation of each step and their corresponding role are described as follows:

2.2.1. Data preparation

The temporal concatenation for ERP data [42] was applied to the ERP data from individual subjects. Concatenating was employed along with the conditions for each individual subject of the group. Hence, given N time points from each condition and F scalp electrode (i.e., each condition data size is $F \times N$), the temporal concatenated data is the size of $F \times (N \times C)$ for each individual where C denotes the number of conditions. Then group-wise averaging was performed to be used in clustering analysis. Thus, the temporal concatenated grand average ERP dataset (from two conditions) is the size of 65×600 for the real data, and for the simulated data (from two conditions) is the size of 28×512. Fig. 2**A** demonstrates the temporal concatenating for subjects and the grand average calculation prepared for feeding individual clustering methods. Less formally, the samples for cluster analysis are the time points, and the primary features are the recorded voltage from the scalp electrodes. In order to assess the proposed method, an additional white Gaussian noise (e.g., 20, 10, 5, 0, −5 dB) as a whole (all electrodes' data) is applied to the prepared grand averaged ERP using the MATLAB function *awgn*. This will be in contrast to the assumption that averaging signal from trials/subjects removes noise from the signal for carrying the most powerful ERP responses.

2.2.2. Consensus clustering

The most popular clustering methods for neuroimaging were employed aimed to initialize consensus clustering, including polarity-independent, i.e., after a polarity adjustment to avoid the risk of putting samples with different polarity in the same cluster, and polarity-invariant methods. The clustering methods for the generation phase were selected to provide an appropriate consensus clustering configuration from our toolbox [33], employing the M-N plot method [1]. This approach selects the clustering methods in which the inner-similarity and duration of the obtained time window for a given ERP are appropriate from various clustering options (e.g., repetitive runs on 2 to 20 clusters). Therefore, for the simulated data, k-means [48] and hierarchical clustering [60] with correlation similarity function, spectral clustering [43] with k-means with Euclidean similarity, and modified k-means [46] were selected. Likewise, for the real data, k-means and hierarchical clustering with correlation similarity, fuzzy c-means (FCM) [6], self-organizing maps (SOM) [24], spectral clustering, and modified k-means were selected.

Moreover, the optimal number of clusters was determined by the

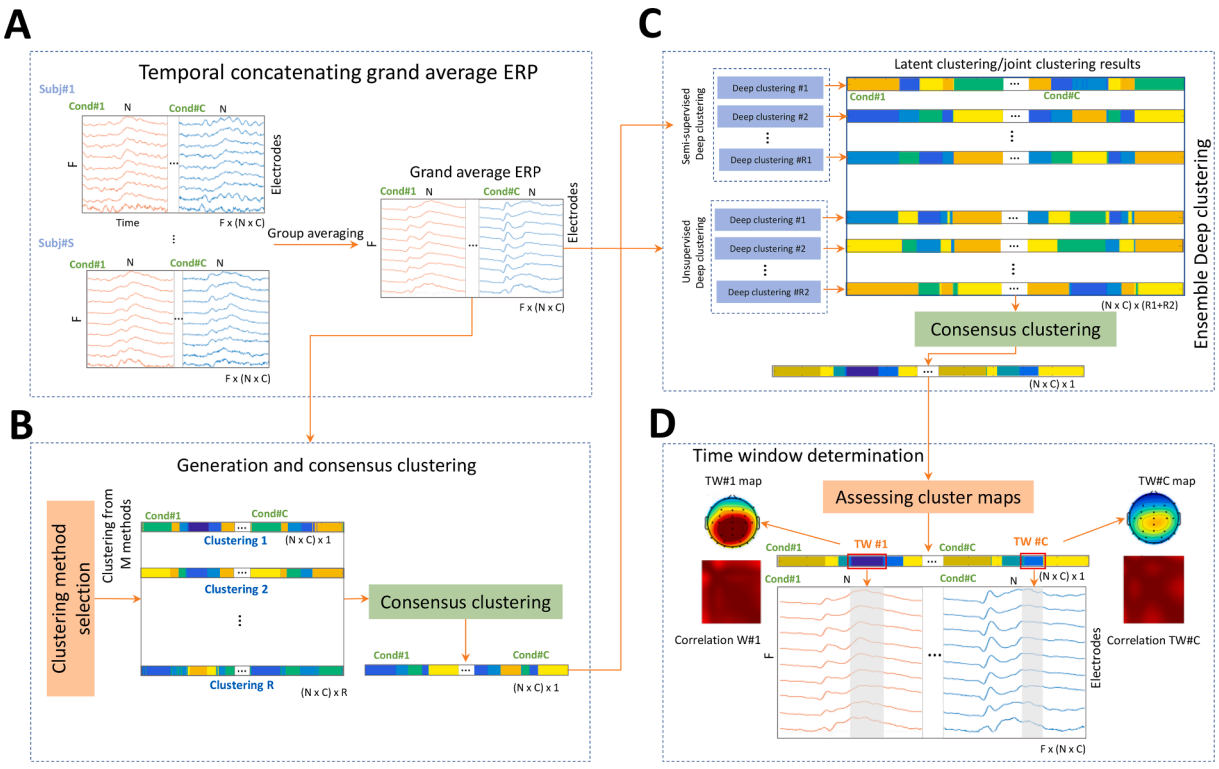

Fig. 2. Proposed pipeline for determining the time window (TW) of the event-related potential (ERP) of interest. **A**. Temporal concatenating data from the C conditions for S subjects and calculating grand average ERP dataset size of $F \times (N \times C)$, where F is the number of electrodes and N denotes the number of time points for each condition dataset. **B**. Selection of the clustering algorithm and generation phase of consensus clustering to initialize semi-supervised deep clustering methods. **C**. Ensemble deep clustering from $R1$ semi-supervised and $R2$ supervised deep clusterings. **D**. Time window determination from the ensemble clustering result. Subj = subject and Cond = condition.

pipeline following our prior work [33] in which the independent repetitive consensus clustering (e.g., up to 100 runs) is performed on the clustering options (e.g., from 2 to 15 clusters) and the inner-similarity of identified time windows is calculated. Then, the optimal number of clusters is estimated in a clustering option where the qualified time windows reach a high inner-similarity (e.g., > 0.95) and become stable. The inner-similarity of a cluster map can be defined as the mean of correlation coefficients between topographical maps for each of two different time points (except self-correlation). As a result, the optimal number of clusters for the simulated and real data were obtained in 6 and 5 clusters, respectively. Hence, once the clustering results are obtained, we apply cluster-based similarity partitioning (CSPA) that is based on hypergraph clustering [57]to calculate the final clustering result.

Mathematically, let us consider the clustering problem of N samples, $X = \{x_1, x_2, ..., x_N\}$ into K groups, where each group is represented by a centroid μ_k, $k = \{1, 2, \cdots, K\}$ and $x_t \in \mathbb{R}^F$, $t = \{1, 2, \cdots, N\}$ and F denotes the number of features, i.e., the electrodes in the EEG scalp. A set of R clusterings $L^{(1,2,\cdots,R)}$ is used for combining into a result clustering L. Therefore, the objective function for cluster ensemble from R clusterings, a consensus function Γ can be defined as:

$$\Gamma : \left\{L^{(i)} \middle| i \in \{1, 2, \cdots, R\}\right\} \rightarrow L \tag{1}$$

which is a function of $\mathbb{N}^{N\times R} \rightarrow \mathbb{N}^N$ that maps clusterings to a final set of clusters. Given a set of clusterings $\{L^{(i)} | i \in \{1, 2, \cdots, R\}\}$, the goal is to explore the clustering result that shares the most information from all clusterings. The mutual information between two clustering results like L_i, L_j is denoted by $I(L_i,L_j)$,and $H(L_i)$ denotes the entropy of L_i. Hence, the normalized mutual information (NMI), i.e., in the range between 0 and 1, between L_i, L_j using geometric mean can be denoted by:

$$NMI(L_i, L_j) = \frac{I(L_i, L_j)}{\sqrt{H(L_i)H(L_j)}} \tag{2}$$

$$I(L_i, L_j) \leq \min\left(H(L_i)H(L_j)\right), \tag{3}$$

$$H(\widehat{L}) = \sum_{a=1}^{K} N_a \log \frac{N_a}{N} \tag{4}$$

where N_a refers to the number of samples in the cluster C_a according to $\widehat{L}$. Thus, for two clustering results L_i, L_j, the mutual information is calculated as:

$$\Gamma^{(NMI)}(L_i, L_j) = \frac{\sum_{a=1}^{K}\sum_{b=1}^{K} N_{a,b} \log\left(\frac{N.N_{a,b}}{N_a N_b}\right)}{\sqrt{\left(\sum_{a=1}^{K} N_a \log\frac{N_a}{N}\right)\left(\sum_{b=1}^{K} N_b \log\frac{N_b}{N}\right)}} \tag{5}$$

where N_a, N_b present the number of samples in the clusters C_a, C_b according to L_i, L_j, respectively. $N_{a,b}$ refers to the number of samples in cluster a according to C_a as well as in cluster b according to C_b. Thus, the mutual information between r clusterings (Λ) can be defined as the average NMI (ANMI):

$$\Gamma^{(ANMI)}(\Lambda, \widehat{L}) = \frac{1}{R}\sum_{l=1}^{R} \Gamma^{(NMI)}(\widehat{L}, L_l) \tag{6}$$

Therefore, the optimal labeling from r clusterings can be simply defined as:

$$L^* = argmax_{L\in\mathbb{L}} \sum_{l=1}^{R} \Gamma^{(NMI)}(L_l) \tag{7}$$

where Γ denotes a similarity measurement (e.g., NMI), which measures mutual information between a set of R clusterings and L^* is an optimal

combined clustering with maximum average similarity to all other clusterings L_l. Note that the L^* (consensus clustering labeling) has the same size with individual labelings L_l. Notably, the applied CSPA consensus function can calculate the clustering result from non-heterogeneous labeling (i.e., different number of clusters or including missing labels) [57]. As a result, once the clustering labels are assigned via clustering methods (generation phase), the consensus function explores the maximum aggregation between the clusterings.

2.2.3. Deep clustering

Two groups of deep clustering methods were designed. First, the semi-supervised methods, i.e., initialized with a consensus clustering result, were applied to the prepared ERP data (i.e., including additional noise) to obtain cluster-friendly transformed data by learning the K class of clusters. The second group was designed based on the end-to-end autoencoder (AE)-based unsupervised deep clusterings to learn the most powerful features of the data for clustering into K groups. Depending on the deep clustering design, the clustering module was embedded as a layer or linked to be fed by the transformed dataset independently. The following describes general mathematical logic for both groups of designed deep clustering methods.

Let X be the prepared data, e.g., size of N × F from N time points and F electrodes, and $Y = \{y_1, y_2, \ldots, y_N\}$ denotes the labels obtained via consensus clustering in the dataset. The transforming function can be defined as $S_\varnothing : X \rightarrow Y$, which maps each time point $x_t = \{e_1, e_2, \cdots, e_F\}$ (i.e., a topography map) associated with a label y_t, $t \in 1, 2, \cdots, N$, where $\varnothing$ are the learnable parameters by the network. The role of the deep clustering method is to assign the input space to K clusters $L = \{C_1, C_2, \cdots, C_K\}$, where $C_k = \{x_t | y_t = k, \forall t \in 1, 2, \cdots, N\}$. Therefore, X and Y are defined as input and output spaces, in which input space is transformed with a nonlinear mapping $f_\theta : X \rightarrow Z$ where θ are learnable parameters and Z is embedded feature space, $Z \in \mathbb{R}^K$. Then a parametrized classifier such as g_ω is used to predict the correct labels on top of the features $f_\theta(x_t)$, where the classifier and mapping parameters ω and θ are jointly learned by optimizing the following problem:

$$min_{\theta,\omega} \frac{1}{N} \sum_{t=1}^{N} Loss_{net}(g_\omega(f_\theta(x_t)), y_t) \tag{8}$$

where $Loss_{net}$ is the multinomial logistic loss, also known as the negative log-*softmax* function.

Regardless of the type of applied layers in the semi-supervised methods, the DNN is encouraged to minimize $Loss_{net}$ in order to optimize the prediction of labels (see Eq. (8)). Hence, the input for semi-supervised deep clustering methods is the prepared grand average ERP data order of $\mathbb{R}^{F \times (N \times C)}$, and the output is the order of $\mathbb{N}^K$ (K notes). Thus, the cluster-friendly transformed data (after training) is the size of $(N \times C) \times K$. Next, we apply a stabilized clustering [31] using k-means for fine-tuning and obtaining the clustering result.

For unsupervised deep clustering methods, the DNN optimizes the network knowledge about input signal jointly with a clustering module, i.e., depending on the design, the clustering module can be connected to the bottleneck layer. The *Loss* function usually is the combination of the network and clustering losses, denoted as follow:

$$Loss = Loss_{net} + \gamma Loss_{cl}, \tag{9}$$

where $Loss_{cl}$ denotes the clustering (embedded) loss. γ is a hyper-parameter, which is used to balance the two costs in jointly learning deep clustering method. Note that $Loss_{net}$ for unsupervised methods is defined depending on the learning method and DNNs' structure. Therefore, the reconstruction loss can be easily defined as:

$$min_{\theta_1,\omega_1} L_{rec} = min \frac{1}{N} \sum_{t=1}^{N} \| x_t - g_{\omega_1}(f_{\theta_1}(x_t)) \|^2 \tag{10}$$

where the network is composed of two groups of layers corresponding to the encoder $f_{\theta_1}(.)$ and decoder $g_{\omega_1}()$ with a bottleneck layer(s). The input of the connected clustering module is the encoder's output from the bottleneck layer as the cluster-friendly data. In this definition, the transformed data size for clustering is $(N \times C) \times K$.

2.2.3.1. Design of studied deep clustering methods. The configuration problem of consensus clustering is considered as finding a balance between the selected clustering methods (i.e., called exploration in machine learning) to obtain optimal/sub-optimal combination (i.e., exploitation). In this study, we provided the M-N plot [1 33] to pre-test (see Fig. 3) the studied methods against the different datasets (regarding the noise levels) to avoid trivial results from the individual methods. As a result, the clustering method with a higher risk of obtaining worse candidate cluster maps (cluster maps in the critical area) with an insufficient number of time points and unstable (considering noise in the data) can be eliminated. Although some of the methods achieved less

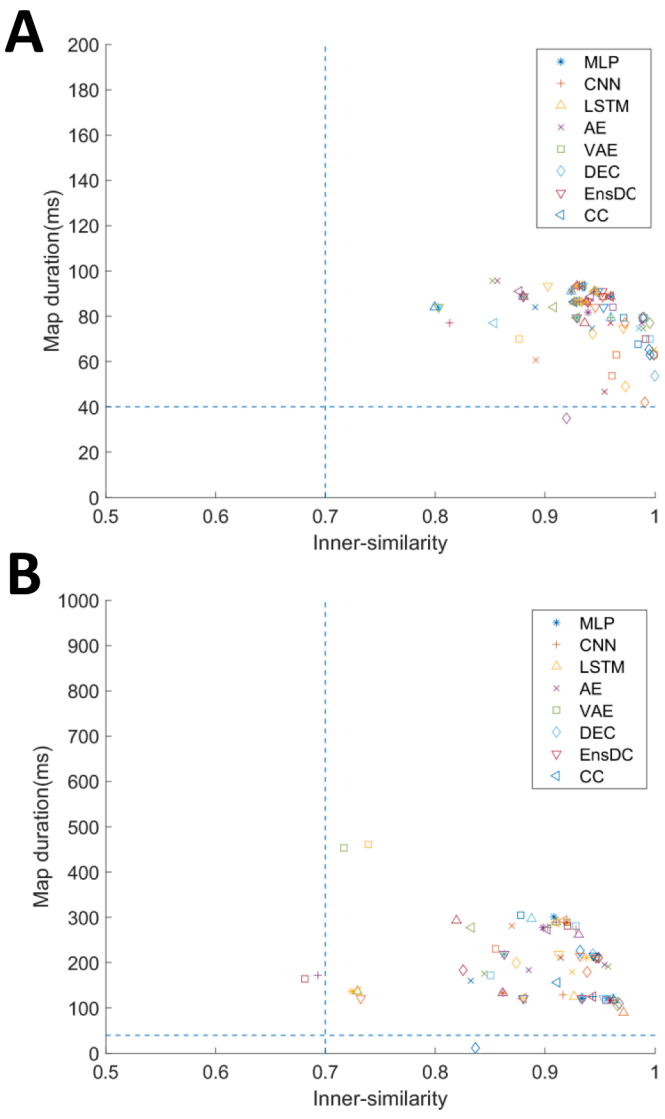

Fig. 3. Illustration of the M–N plot pre-test of the studied methods on the prepared simulated data (**A**) and the real data (**B**) with the additional noises. All the studied deep clustering designs identified cluster maps with high inner similarity (particularly noisy data resulted in lower inner similarity) from both ER data. MLP = fully connected multi-layer perceptron, CNN = one dimensional convolutional deep neural network, LSTM = long short-term memory, AE = autoencoder, VAE = variational autoencoder, DEC = deep embedded clustering, EnsDC = ensemble deep clustering, and CC = consensus clustering.

compact or even fewer samples in candidate cluster maps, i.e., in some of the tests, this does not affect the proposed method's performance as there are no trivial results from the proposed method (Ens_DC) in Fig. 3**A** and Fig. 3**B**. This can also estimate whether those clustering methods are appropriate for given ERP data with additional noise.

We designed six standard deep clustering models from the popular deep clustering designs for sequential data after evaluating them. Therefore, from the semi-supervised methods, we designed: a fully-connected multilevel perceptron (FC-MLP) [39] DNN to learn the essential features of ERP data; a one-dimensional convolutional neural network (1D CNN) [9] to learn complex features of the prepared ERP data; and long short-term memory (LSTM) [26] to learn sequential features of data. We employed a stabilized clustering module via consensus clustering [31] to cluster the transformed data from each DNN.

For the unsupervised group, we designed: an end-to-end AE deep network [40] to use the capability of AE for learning ERP features; variational autoencoder (VAE) [22] for learning input space distribution (e.g., Gaussian distribution) in the latent space; and deep embedded clustering (DEC) [12] to simultaneously learn feature representations and optimize cluster assignments (soft assignment). Likewise, in the first group, a stabilized clustering module was used for clustering transformed data from the encoder's output of AE and VAE based deep clustering methods. Table 1 illustrates the designed blocks of deep clustering models for prepared ERP datasets.

For the DNNs' hyperparameters, we have used the sparse categorical cross-entropy loss function as the net loss and a common *adam* optimizer for supervised and *RMSProp* optimizer for the semi-supervised group with default hyperparameters. Furthermore, the other hyperparameters of all the DNNs models, such as the number of units, the number of layers, the learning rate (e.g., 0.001), batch size (e.g., 150), and the number of iterations (e.g., 100 iterations) were determined by tuning the network using a coarse grid search [5]. Finally, 5-fold cross-validation with mentioned optimizers has been applied to 80 percent of the data for training and validation and 20 percent for the test evaluation. All DNNs were built-in using Keras deep learning libraries.

2.2.3.2. Ensemble deep clustering. Among different strategies for ensemble clustering [8,54], we combined the results of individual methods to calculate the ensemble result from non-heterogeneous elements (i.e., various deep DNNs strategies). Once the results from deep clustering models are obtained, the deep clustering results are fed into consensus clustering (CSPA consensus function) for exploring the most aggregate clustering result. Hence, the consensus clustering at the deep clustering level combines the labeling results from semi-supervised deep clustering methods, AE and VAE from the supervised group, and the clustering result of labeling optimization from DEC (see Fig. 2**C**).

Mathematically, following the same principle to calculate the mutual information in Eq. (7) (in Section 2.2.2), the ensemble clustering can be described as:

$$\widehat{L}^{*} = argmax_{L\in\mathbb{L}} \sum_{l=1}^{\hat{R}} \Gamma(\widehat{L}_l) \tag{11}$$

where $\widehat{L}^{*}$ is the result of consensus clustering from $\widehat{R} = R1 + R2$ deep clustering methods (i.e., including $R1$semi-supervised clustering and $R2$ unsupervised deep clustering methods) and $\widehat{L}_l$ represents the results from all deep clustering methods. We used the CSPA consensus function, which has suitable tolerance for selecting the number of clusters and the combination of unstable clusters [57]. Therefore, the labeling result from the mentioned three semi-supervised and three unsupervised deep clusterings (i.e., size of 600×6 for the simulated data and 512×5 for

Table 1
Illustration of designed deep clustering models applied to ERP datasets where N is the number of time points, F is the number of electrodes, and C is the number of conditions. FC_MLP = fully connected multi-layer perceptron, 1D_CNN = one dimensional convolutional deep neural network, LSTM = long short-term memory, AE = autoencoder, VAE = variational autoencoder, DEC = deep embedded clustering, Relu = rectified linear unit, tanh = hyperbolic tangent function, LB = consensus clustering results for feeding the semi-supervised methods. P = current estimation of clustering labels, Q = the previous estimation of the labels, and KL = Kullback-Leibler divergence.

	Semi-supervised			**Unsupervised**		
Deep clustering	**FC_MLP**	**1D_CNN**	**LSTM**	**AE**	**VAE**	**DEC**
Input	(N × C) × F, LB	(N × C) × F, LB	(N × C) × F, LB	(N × C) × F	(N × C) × F	(N × C) × F
Layer 1	FC (64, Relu)	1D_Conv (64, Relu, input), kernel = 1	Lstm (64, Relu)	FC (256, tanh)	FC (125, tanh)	FC (64, Relu)
Layer 2	Batch normalization	1D_Conv (64, Relu), kernel = 1	Batch normalization	Batch normalization	Batch normalization	Batch normalization
Layer 3	Dropout 5%	Max_Pooling_1D	Dropout 5%	Dropout 2%	Dropout 5%	Dropout 5%
Layer 4	FC (512, Relu)	Batch normalization	Lstm (128, Relu)	FC (512, tanh)	FC (256, tanh)	FC (256, Relu)
Layer 5	Batch normalization	Dropout 5%	Batch normalization	Batch normalization	Batch normalization	Batch normalization
Layer 6	Dropout 5%	1D_Conv (256, Relu), kernel = 1	Dropout 5%	Dropout 5%	Dropout 5%	Dropout 5%
Layer 7	FC (256, Relu)	1D_Conv (256, Relu), kernel = 1	FC (128, Relu)	FC (K, Softmax)	FC (256, tanh)	FC (256, tanh)
Layer 8	Batch normalization	Batch normalization	Batch normalization	**Clustering**	Batch normalization	Batch normalization
Layer 9	Dropout 5%	Dropout 5%	Dropout 5%	FC (512 tanh)	Dropout 5%	Dropout 5%
Layer 10	FC (128, Relu)	1D_Conv (64, Relu), kernel = 1	FC (64, Relu)	Batch normalization	Z(z_mean(K), z_log_var(K))	**Clustering Layer (KL-divergence (P,Q))**
Layer 11	Batch normalization	1D_Conv (64, Relu), kernel = 1	Batch normalization	Dropout 5%	Lambda (sampling)	FC (256, Relu)
Layer 12	Dropout 5%	Global_average_pooling_1D	Dropout 5%	FC (256, tanh)	**Clustering**	Batch normalization
Layer 13	FC (K, Softmax)	Batch normalization	FC (K, Softmax)	Batch normalization	FC (256, tanh)	Dropout 5%
Layer 14	**Clustering**	FC (K, Softmax)	**Clustering**	Dropout 5%	Batch normalization	FC (256, Relu)
Layer 15		**Clustering**			Dropout 5%	Batch normalization
Layer 16					FC (256, tanh)	Dropout 5%
Layer 17					Batch normalization	FC (64, Relu)
Layer 18					Dropout 5%	Batch normalization
Layer 19					FC (128, tanh)	Dropout 5%
Layer 20					Batch normalization	
Layer 21					Dropout 5%	

the real data) was achieved in a firm aggregate labeling of 600 and 512 time points in the concatenated simulated and the real ERP data, respectively.

2.2.4. Time window determination

We modified the previously designed time window determination method [31] to provide more flexibility in the inner similarity and duration thresholds of candidate cluster maps. It should be noted that the time window determination method requires experimental information about the ERP of interest. This means that considering the experimental design (e.g., visual and auditory) and participant group (e.g., age, sex, and health level), a rough expectation (at least based on stimulation) of some neurological brain response (e.g., attention, memory, and mismatch components) is approachable. Therefore, the stimulation onset/offset time, the target response, and the electrode site are expected. The adaptive time window adjusts the inner similarity threshold (e.g., $0.7 \leq$ minimum inner-similarity ≤ 0.95) and the consecutive number of time points in the candidate cluster maps (e.g., 30 ms $\leq$ minimum number of time points $\leq$ 50 ms) while needed. In other words, the time window determination method starts from the highest possible inner similarity with sufficient duration and applies a silent change (e.g., 0.003 for inner similarity and 2 ms for the duration of the map, which can be adjusted when needed) if no suitable representative map is found.

2.3. Performance analysis

2.3.1. Evaluation metrics

We applied the popular performance evaluation metrics, namely, accuracy (ACC) [65], NMI [57], and adjusted rand index (ARI) [37], to assess the performance of clustering methods. Hence, given the known clustering L (i.e., ground-truth) and the clustering result L', the accuracy index can be defined as:

$$ACC(L, L') = max \frac{\sum_{i=1}^{N} 1\{L(i) = m(L'(i))\}}{N}, \tag{12}$$

where m provides overall possible one-to-one mappings between clusters and labels using the Hungarian algorithm [25]. It is, however, not possible to get the same label for the given similar clusters from multiple clustering methods (i.e., different labels might be generated for the same cluster in multiple runs or various methods). Therefore, the Rand index (RI), as a suitable index to compare clustering results, can be defined as:

$$\mathscr{R}(L, L') = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

where TP, TN, FP, and FN signify true positive, true negative, false-positive, and false-negative rates, respectively. By calculating the expectation of R, i.e., $E[R]$, the adjusted rand index (ARI) is calculated as:

$$ARI(L, L') = \frac{R(L, L') - E[\mathscr{R}]}{1 - E[\mathscr{R}]}. \tag{14}$$

Besides, mutual information provides a suitable concept of the shared information between a pair of clusterings as an asymmetric measure to quantify the statistical information shared between two distributions [59], which we have defined in subsection 2.2.2. Another reasonable index called adjusted mutual information (AMI) for NMI [63] is used by calculating the NMI expectation as the following:

$$AMI(L, L') = \frac{I(L, L') - E\{I(L, L')\}}{\max\{H(L), H(L')\} - E\{I(L, L')\}} \tag{15}$$

The ground truth clustering for the comparison using those metrics mentioned above is the clustering results on the prepared grand average ERP data using state-of-the-art consensus clustering when no additional noise is applied. The clustering performance of the studied clustering methods is assessed from the data of the different additional noise.

2.3.2. Statistical analysis

We provided a standard analysis of variances to determine whether the identified P3 effect (from each ERP data) is statistically significant. For the simulated data, statistical analysis was carried out via a repeated measures analysis of variance (rmANOVA) with a within-subject factor: *Task* (conditions: 'Cond1′ and 'Cond2′) in two pre-defined electrode sites CPz and Cz. This was performed by measuring the mean voltage of the P3 amplitude in the determined time window. The effect of the *Task* was tested against the null hypothesis of existing no significant difference between the conditions from those selected electrodes and the estimated time windows. Likewise, statistical power analysis for real data was carried out via a rmANOVA with a within-subject factor: *Task* (conditions: 'Rare' and 'Frequent') by measuring the mean amplitude of P3 on the priory selected electrode site over the parietal region (electrode: Pz). We tested the effect of the *Task* for the hypothesis that a significant difference exists between the 'Rare' and 'Frequent' conditions at the selected electrode site and time windows. Statistical comparisons were made at p-values of $p < 0.05$ for both data.

3. Results and evaluations

The summarized results of applying the proposed pipeline to two ERP data are illustrated, including the performance of each DNN, clustering results, estimated time windows, and statistical analysis results from different noise levels. Furthermore, we present the performance results based on the defined metrics.

3.1. Performance of the studied DNNs

Table 2 and Table 3 show the training performance of the studied DNNs on the test datasets from the simulated and real data, respectively. Observing the results in Table 2 and Table 3 discloses that for semi-supervised DNNs, the DNNs are able to learn the ERP data and the labeling depending on the DNNs' structures with high accuracy, even for noisy data. The unsupervised DNN models, on the other hand, have learned the input space properties with relatively worse *Loss* rates than the semi-supervised methods. Together, from the results of both datasets, the designed DNNs successfully trained on the prepared ERP data from additional noises.

3.2. Clustering results and time windows

Fig. 4 shows the clustering results (randomly selected) from the proposed and state-of-the-art methods (consensus clustering) in the simulated data, i.e., when no additional noise exists and the maximum reasonable noise is added (e.g., −5 dB). We excluded results from datasets with additive noise between them to keep the figure readable. The qualified cluster maps for identifying the interesting ERP were marked in gray color for both Fig. 4 and Fig. 5. Observing Fig. 4**A,** i.e., results of consensus clustering in the prepared simulated data without noise, shows that the time windows for the P3 component are isolated with cluster maps 5 (colored gray areas) from 268.67 to 355.00 ms for 'Cond1′ and 268.67 to 362.00 ms (ground-truth) for 'Cond2′. Similarly, in Fig. 4**B** (i.e., the proposed method results), those time windows have been elicited by maps 4 in the identical time windows, i.e., in 268.67 to 355.00 ms and 268.67 to 362.00 ms for 'Cond1′ and 'Cond2′, respectively. P3 was isolated for highly noisy simulated data by maps 6 from 273.33 to 350.33 ms for 'Cond1′ and 271.00 to 355.00 ms for 'Cond2′ using consensus clustering (see Fig. 4**C**). The proposed method extracted P3 by maps 4 from 273.33 to 352.67 ms and 268.67 to 357.33 ms for 'Cond1′ and 'Cond2′ (see Fig. 4**D**) for the noisy simulated data. Noticeably, a larger peak was observed (in the determined time windows) in 'Cond1′ than in 'Cond2′ when no noise was added and from the maxima noisy datasets, from the clustering results via two methods.

Observing Fig. 5 (a randomly selected result), for the real data with no additional noise, P3 was isolated by map 1 and map 2 from 303.90 to

Table 2
The studied DNNs' performances (on the test dataset) in the simulated data while additional noise is included on the original signal (i.e., from 20 dB to −5 dB). acc = accuracy, SNR = signal-to-noise ratio. The SNR value denotes the additive white Gaussian noise in the prepared ERP signal.

Method	No noise added		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5 dB	
	loss	acc	loss	acc	loss	acc	loss	acc	loss	acc	loss	acc
FC_MLP	0.002	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
1D CNN	0.002	1.000	0.000	1.000	0.000	1.000	0.001	1.000	0.000	1.000	0.001	1.000
LSTM	0.002	1.000	0.003	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
AE	0.002	–	0.004	–	0.003	–	0.004	–	0.003	–	0.003	–
VAE	0.016	–	0.011	–	0.032	–	0.037	–	0.040	–	0.042	–
DEC	0.046	–	0.056	–	0.050	–	0.050	–	0.053	–	0.060	–

Table 3
The studied DNNs' performances on the test dataset for the real data when the additive noise increases from 20 dB to −5 dB.

Method	No noise added		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5 dB	
	loss	acc	loss	acc	loss	acc	loss	acc	loss	acc	loss	acc
FC_MLP	0.003	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
1D CNN	0.001	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
LSTM	0.004	1.000	0.001	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
AE	0.012		0.015	–	0.016	–	0.021		0.026		0.029	–
VAE	0.022	–	0.035	–	0.035	–	0.044		0.040		0.050	–
DEC	0.016	–	0.024	–	0.019	–	0.034		0.042		0.040	–

514.84 ms and 342.97 to 464.06 ms (ground-truth), in condition (1) and condition (2), respectively, using the consensus clustering (Fig. 5**A**). Those time windows for the P3 component were elicited by map 1 and map 2 (Fig. 5**B**), in 303.90 to 514.84 ms and 342.97 to 460.16 ms for conditions (1) and (2), respectively, using the proposed method. While a high noise (SNR = -5 dB) was added in the real data, P3 was isolated by maps 1 from 331.25 to 589.06 ms and 342.97 to 428.91 ms in conditions (1) and (2), respectively, using consensus clustering. Whereas those time windows were elicited by map 1 and map 2 (colored gray areas), from 296.09 to 514.84 ms and 354.69 to 475.78 ms, respectively, using the proposed method. Together, the clustering results for both ERP data (the simulated and real) with different amounts of additive noise revealed that the clusterings include noisier clusters, particularly where no strong response exists (e.g., pre-stimulus onset). Observably, the proposed method seems to provide a more robust clustering result (Fig. 4**D** and Fig. 5**D**) than consensus clustering (Fig. 4**C** and Fig. 5**C**).

We provided detailed results of the estimated time windows (start, end, and duration) of the P3 identification from the studied method in different noise levels in Table 4 and Table 5**.** For simulated data (see Table 4), all the studied methods identified P3 response with some degree of accuracy. However, consensus clustering obtained better time window accuracy, especially in real data when adding non-intensive noise. Semi-supervised deep clustering obtained suitable identification due to supervising by consensus clustering. The proposed method obtained more accurate and stable results among different methods. Likewise, for the real data (see Table 5), the time window of the P3 response was identified from the clustering results of all the methods studied. Notably, the proposed method achieved more accuracy and stability than other studied methods.

Furthermore, Table 6 illustrates the standard deviation error of the estimated time windows from the different clustering methods in the examined datasets with additional noises. Together, the time window determination and the stability evaluation results reveal that all the deep clustering methods successfully identified P3 from different datasets. Our method performed better in the real data than in the simulated data, which was relatively better than other methods.

To provide more evidential results and test the spatial properties of qualifying the isolated ERPs of interest, we examined the spatial correlation between the mean topography maps in the ground-truth time window and the time windows from the studied methods. We included the spatial correlation test results in Table 7 and Table 8 in the simulated and the real data, respectively. The results showed a high spatial correlation between the identified ERP and the ground truth P3 topographical map from the majority of the studied methods. However, a silently less correlation was obtained in low SNR due to the noise effect on topography configuration.

3.3. Evaluation and statistical analysis results

Fig. 6**A** and Fig. 6**B** show the clustering performance based on the performance metrics (ACC, ARI, and AMI) for simulated and real data, respectively. We have included the performance of the studied deep clustering methods for a better comparison and understanding of their role in the proposed clustering. Observing Fig. 6 exposes a suitable performance from the studied deep clustering methods and, consequently, the ensemble deep clustering (proposed method). On the other hand, consensus clustering results reveal a suitable performance with comparatively better stability while the noise ratio is changed. Observing Fig. 6**A** indicates that, except for the clean data, the proposed method provides a more confident performance than the studied clustering methods, especially consensus clustering in the simulated data. Likewise, Fig. 6**B** shows that the proposed method obtained remarkable and stable results while the data noise varied in the real data. Noticeably, except for the ground truth in the simulated data and corresponding semi-supervised methods' performance, the proposed method discloses a relatively superior and stable performance for both datasets.

For the studied methods, the statistical analysis results of the elicited P3 effects from the estimated time windows (see Table 4 and Table 5) were illustrated in Table 9 and Table 10 for the simulated and real ERP data, respectively. For the simulated data, our results revealed a large P3 effect and a significant difference between the conditions ($F(1,19) = 81317$, p -value < 0.0001, $\eta_p^2 = 1.000$) in the region of the interest (the central area) and the obtained time windows when no additional noise exists. However, the obtained large effect and calculated highly significant difference between the conditions can be seen to be overestimated. This can be because of the occurred alignment in the subjects' responses in the peak/mean amplitude (i.e., due to being calculated in the same ratio with silent changes) from the simulation mechanism when there is no additional noise. Hence, a larger response was identified in 'Cond1′ than in 'Cond2′, which was expected following the simulation mechanism. Observing Table 9 reveals a silent decrease in the obtained effect size while the data are noisier. Nevertheless, regardless of the noise

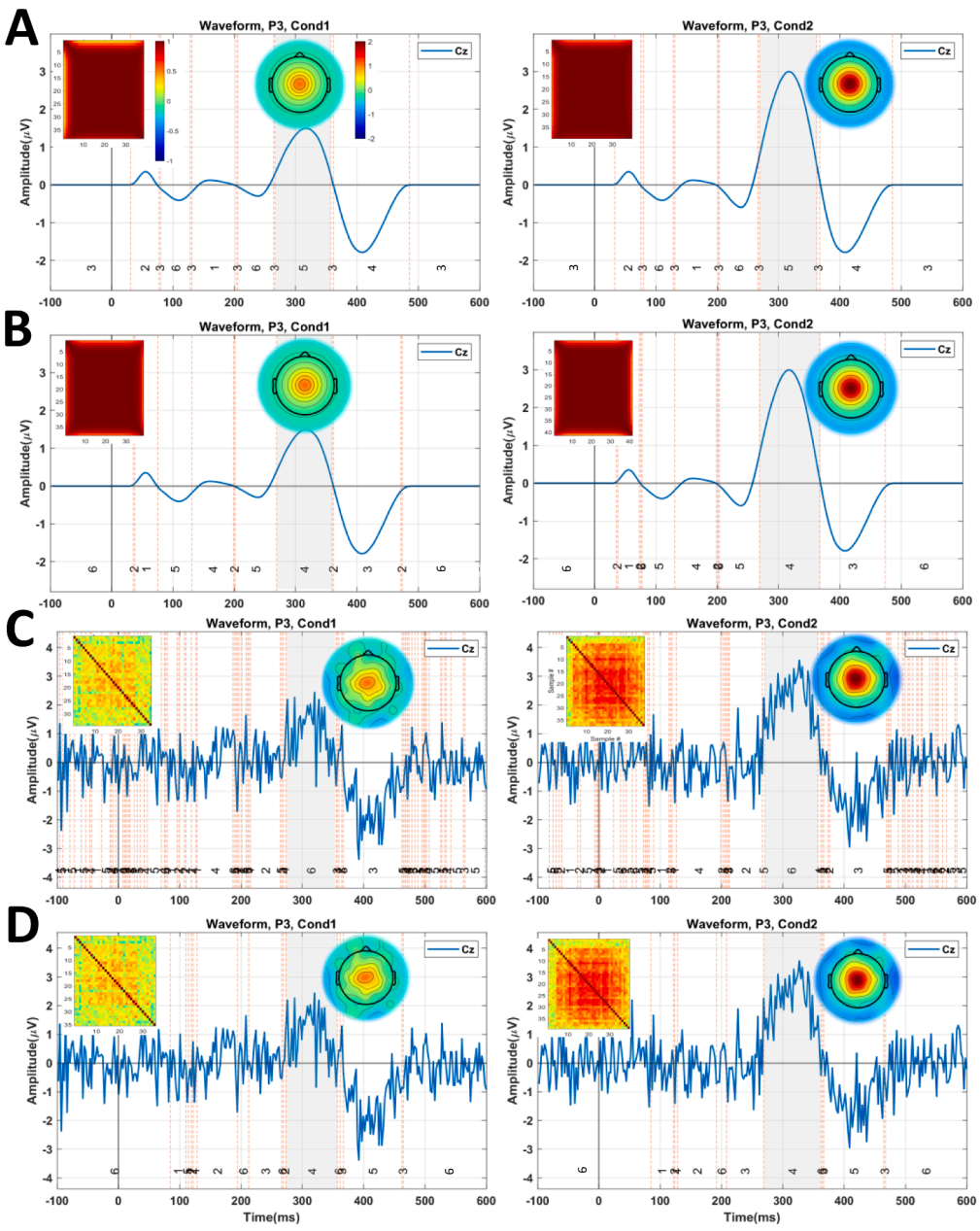

Fig. 4. Illustration of clustering results and selected time windows (colored gray areas), including the corresponding topographies and correlation between the time points for identifying the P3 component in the simulated data. **A.** Isolated time windows with maps 5 (cluster maps 5) in Cond1 and Cond2 from consensus clustering result when no additional noise is added. **B.** Identified time windows by maps 4 in both conditions from the proposed method clustering result when no additional noise is added. **C.** Identified time windows with maps 6 (in both conditions) from consensus clustering in maximum additional noise of −5 dB. **D.** Isolated time windows with maps 4 (in both conditions) of the proposed method when the additional noise is −5 dB. The numbers for each segment present the associated cluster map's number. Cond1 = condition (1), Cond2 = condition (2).

level, a large effect size was obtained from the majority of the studied clustering methods in the estimated time windows.

For real data, our results confirmed the previous findings on the main effect of the *Task*, indicating existing large effect size and significant difference ($F(1,39) = 121.18$, p-value < 0.0001, $\eta_p^2 = 0.76$) between conditions that was larger amplitude in the 'Rare' condition (target) than'Frequent' condition (non-target) in the central lobe of the occipital region. Table 10 shows the results of the statistical power analysis from different noise levels for the studied methods. Similar to the simulated data, all the studied methods identified a significant effect of P3 in the estimated time windows.

4. Discussion

This study presents the ensemble deep clustering pipeline to reliably determine the time window of the ERP of interest when existing noise on the ERP data is unknown after preprocessing. To tackle the problem, we designed the ensemble clusterings from multiple deep clustering methods, including semi-supervised and unsupervised, to explore the most aggregated clusters in the data. To this end, we clustered the weighted data from the trained DNNs (in the latent space), namely, FC_MLP, LSTM, 1DCNN, AE, and VAE, for fine-tuning. Then, those clusterings and DEC's results were combined using the consensus function to calculate the final clustering result. Finally, the modified time window determination was used for estimating the ERP of interest from the candidate cluster maps in each condition/group. The proposed pipeline was built on three methodologies, cluster analysis of spatio-temporal ERP, deep learning as a powerful and noise-tolerant tool, and ensemble learning.

The idea of ensemble clustering in this study is that the scalp EEG data recorded from the same or different devices, multiple-subject (e.g., age, brain size, healthy level) from different conditions/groups carries various artifacts even after preprocessing affects the quality of data. On the other hand, considering the fact that even the popular clustering algorithms could fail spectacularly for certain datasets that do not match the corresponding modeling assumptions (Acharya and Ghosh, 2011),

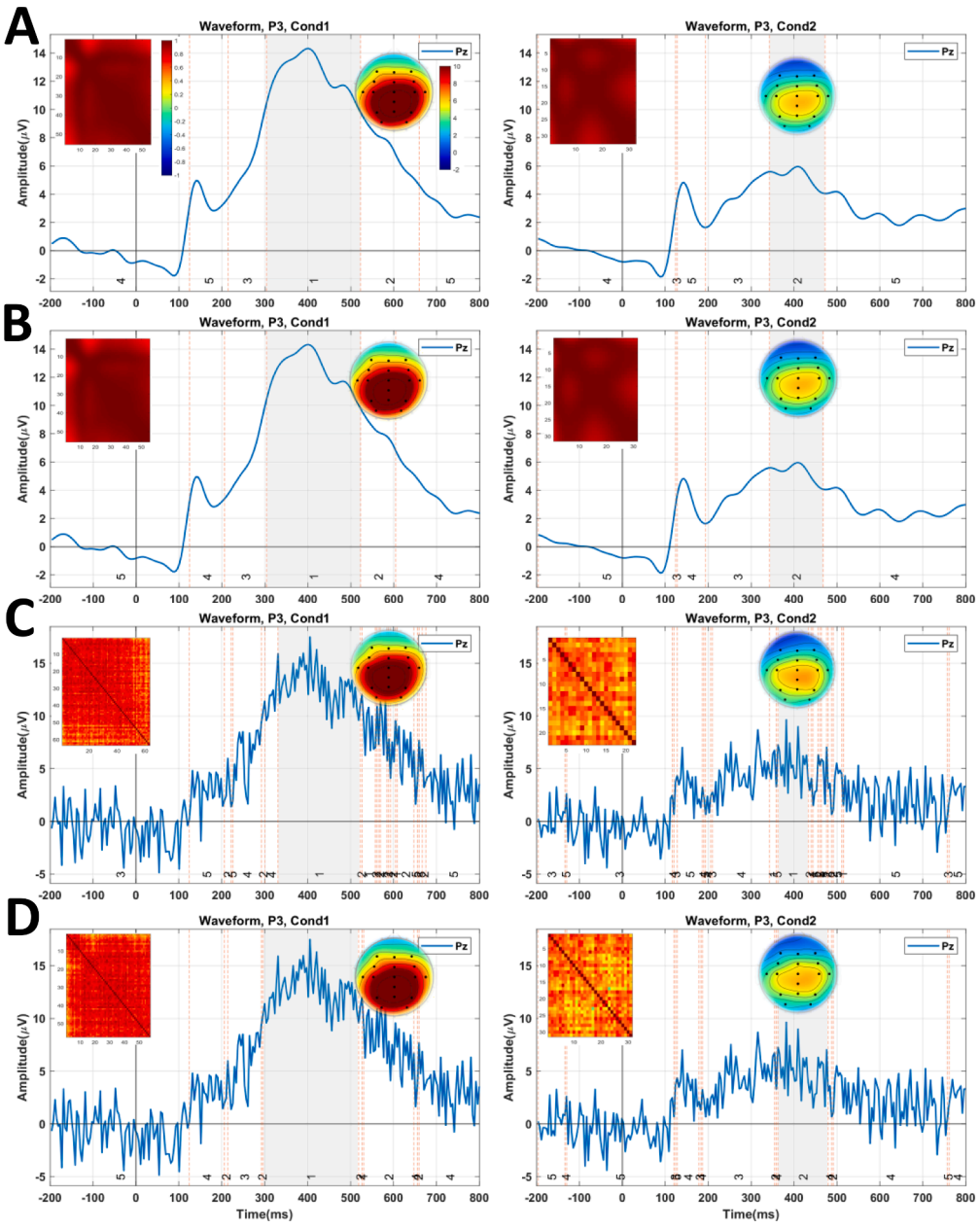

Fig. 5. Illustration of clustering results and selected time windows (colored gray areas) from the proposed and consensus clustering methods for identifying the P3 component in the real data. The results of each condition include the corresponding topographies and the correlation between the time points for the determined time window. **A.** Isolated time windows by maps 1 and 2 for Cond1 and Cond2, respectively, using consensus clustering when no additional noise is applied. **B.** Identified time windows by maps 1 and 2 in Cond1 and Cond2, respectively, using the proposed method when no additional noise is applied. **C.** Identified time windows with maps 1 (in both conditions) by consensus clustering in maximum additional noise of −5 dB. **D.** Isolated time windows with maps 1 and 2 Cond1 and Cond2 by the proposed method when the additional noise is −5 dB, respectively.

using a more reliable clustering method is suggested. In addition, clustering noisy or unbalanced EEG/ERP tensors considering the only spatial properties can result in unreliable cluster maps (i.e., for qualifying the ERP components) since numerous small peaks can be recognized as brain responses [38]. As a result, although available clustering techniques besides ICA/PCA provided a more reliable decomposition of ERP of interest, more challenging data can lead to a problematic result (e.g., determination of divided component, missing ERP). These problems can be more severe if inappropriate preprocessing is performed.

One important issue with ensemble learning methods is the configuration consistency for such a combination. Although this mechanism eliminates the contribution of trivial results, it cannot guarantee the optimization of ensemble clustering. Considering the fact that there is no straightforward solution for the configuration of ensemble clustering [61] and existing a large variety of deep clustering designs [51], we provided an M-N plot pre-test of the studied methods against the different datasets (in terms of noise level) to avoid using deep clustering methods with trivial results. Noting that we avoided testing sophisticated deep clustering designs to keep our design implementable and understandable at this stage. Our early findings revealed three important characteristics of using DNNs for training ERP data. First, the studied DNNs are powerful learners in learning ERP data even when data is considerably noisy. Next, the studied individual deep clustering methods result in clustering in which the interesting components in two datasets and other few components (e.g., N4 in the simulated data) can be identified using the time window determination method. Finally, the ensemble deep clustering provides stable performance compared to other methods associated with the proposed ensemble deep clustering tolerance to artifacts (particularly with noise) without compromising the performance.

The advantages of the proposed method compared to conventional methods are: **i)** using the minimum amount of knowledge in the designed deep clustering methods; **ii)** exploring a firm clustering model for spatio-temporal ERP data by ensemble deep clustering results; **iii)** designing the adaptive time window determination, considering the spatial and temporal properties, from noisy data; **iv)** obtaining the

Table 4
The temporal properties of the estimated time windows (start, end, and duration) through the proposed method, the consensus clustering, and the studied deep clustering methods to qualify the P3 component in the prepared simulated data with different additive noises. The bold marks represent the significant results. Ens_DC = ensemble deep clustering (proposed method), CC = consensus clustering, Cond1 = condition (1), and Cond2 = condition (2).

Method	Properties(ms)	No noise		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5dB	
		Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2
Ens_DC	Start	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	**273.33**	**268.67**	**273.33**	**268.67**
	End	**355.00**	**362.00**	**355.00**	**362.00**	**355.00**	**362.00**	**355.00**	**359.67**	**352.67**	**362.00**	**352.67**	**357.33**
	Duration	**86.33**	**93.33**	**86.33**	**93.33**	**86.33**	**93.33**	**86.33**	**91.00**	**79.33**	**93.33**	79.33	**88.67**
CC	Start	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	273.33	**268.67**	**273.33**	271.00	**273.33**	**268.67**
	End	**355.00**	**362.00**	**355.00**	**362.00**	**355.00**	**362.00**	**355.00**	**359.67**	**352.67**	357.33	350.33	**357.33**
	Duration	**86.33**	**93.33**	**86.33**	**93.33**	**86.33**	**93.33**	81.67	**91.00**	**79.33**	86.33	77.00	**88.67**
MLP_FC	Start	266.33	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	273.33	**268.67**	**273.33**	**268.67**	**273.33**	**268.67**
	End	352.67	357.33	**355.00**	**362.00**	**355.00**	**362.00**	**355.00**	**359.67**	**352.67**	359.67	**357.33**	**357.33**
	Duration	**86.33**	88.67	**86.33**	**93.33**	**86.33**	**93.33**	81.67	**91.00**	**79.33**	91.00	**84.00**	**88.67**
1DCNN	Start	266.33	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	273.33	**268.67**	**273.33**	**268.67**	280.33	**268.67**
	End	352.67	357.33	**355.00**	**362.00**	**355.00**	**362.00**	352.67	**359.67**	**352.67**	359.67	**357.33**	**357.33**
	Duration	**86.33**	88.67	**86.33**	**93.33**	**86.33**	**93.33**	79.33	**91.00**	**79.33**	91.00	77.00	**88.67**
LSTM	Start	266.33	**268.67**	**268.67**	**268.67**	**268.67**	**268.67**	273.33	**268.67**	275.67	**268.67**	**273.33**	**268.67**
	End	352.67	357.33	**355.00**	**362.00**	**355.00**	**362.00**	352.67	**359.67**	**352.67**	359.67	**357.33**	**357.33**
	Duration	**86.33**	88.67	**86.33**	**93.33**	**86.33**	**93.33**	79.33	**91.00**	77.00	91.00	**84.00**	**88.67**
AE	Start	273.33	287.33	273.33	271.00	282.67	**268.67**	273.33	271.00	**273.33**	**268.67**	**273.33**	**268.67**
	End	350.33	341.00	348.00	357.33	343.33	352.67	348.00	355.00	348.00	357.33	345.67	**357.33**
	Duration	77.00	53.67	74.67	86.33	60.67	84.00	74.67	84.00	74.67	88.67	72.33	**88.67**
VAE	Start	280.33	273.33	275.67	273.33	285.00	271.00	278.00	266.33	280.33	273.33	275.67	266.33
	End	343.33	350.33	345.67	352.67	341.00	352.67	345.67	357.33	343.33	352.67	**352.67**	**357.33**
	Duration	63.00	77.00	70.00	79.33	56.00	81.67	67.67	**91.00**	63.00	79.33	77.00	**91.00**
DEC	Start	287.33	275.67	280.33	273.33	287.33	275.67	278.00	**268.67**	290.00	273.33	294.33	278.00
	End	341.00	352.67	343.33	352.67	338.67	350.33	345.67	357.33	341.00	350.33	329.33	350.33
	Duration	53.67	77.00	63.00	79.33	51.33	74.67	67.67	88.67	51.00	77.00	35.00	72.33

Table 5
The temporal properties of estimated time windows via the proposed method, consensus clustering, and the studied deep clustering methods in the real data to qualify the P3 component for different additional noise. The bold marks are the significant results.

Method	Properties(ms)	No noise		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5dB	
		Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2
Ens_DC	Start	**303.91**	**342.97**	**303.91**	339.06	300.00	**342.97**	**300.00**	**346.88**	**303.91**	**342.97**	296.09	354.69
	End	**514.84**	460.16	**514.84**	**460.16**	**514.84**	**464.06**	**518.75**	**467.97**	522.66	**464.06**	**514.84**	**475.78**
	Duration	210.94	117.19	**210.94**	**121.09**	**214.84**	**121.09**	**218.75**	**121.09**	**218.75**	**121.09**	**218.75**	**121.09**
CC	Start	**303.91**	**342.97**	**303.91**	**342.97**	303.91	**342.97**	315.63	**346.88**	**303.91**	**342.97**	331.25	**342.97**
	End	**514.84**	**464.06**	**514.84**	491.41	592.97	**464.06**	592.97	**467.97**	600.78	479.69	589.06	428.91
	Duration	**211.94**	**122.09**	**210.94**	148.44	289.06	**121.09**	277.34	**121.09**	296.88	136.72	257.81	85.94
MLP_FC	Start	292.19	335.16	**303.91**	**342.97**	**303.91**	**342.97**	311.72	**346.88**	331.25	**342.97**	307.81	346.88
	End	596.88	460.16	565.63	444.53	604.69	503.13	581.25	**467.97**	608.59	456.25	510.94	440.63
	Duration	304.69	125.00	261.72	101.56	300.78	160.16	269.53	**121.09**	277.34	113.28	203.13	93.75
1D CNN	Start	292.19	335.16	**303.91**	335.16	**303.91**	339.06	**300.00**	335.16	315.63	**342.97**	288.28	**342.97**
	End	592.97	460.16	565.63	432.81	592.97	**464.06**	592.97	**467.97**	600.78	483.59	510.94	440.63
	Duration	300.78	125.00	261.72	97.66	289.06	125.00	292.97	132.81	285.16	140.63	222.66	97.66
LSTM	Start	296.09	335.16	**303.91**	**342.97**	292.19	339.06	**300.00**	**339.06**	315.63	**342.97**	307.81	**342.97**
	End	592.97	460.16	596.88	460.16	592.97	**464.06**	577.34	**467.97**	608.59	479.69	510.94	440.63
	Duration	296.88	125.00	292.97	117.19	300.78	125.00	277.34	128.91	292.97	136.72	203.13	97.66
AE	Start	307.81	346.88	311.72	**342.97**	300.00	335.16	**300.00**	**339.06**	**303.91**	339.06	307.81	346.88
	End	503.13	487.50	503.13	526.56	**514.84**	510.94	**518.75**	503.13	526.56	460.16	507.03	507.03
	Duration	195.31	140.63	191.41	183.59	**214.84**	175.78	**218.75**	164.06	222.66	**121.09**	199.22	160.16
VAE	Start	284.38	335.16	311.72	331.25	300.00	**342.97**	**300.00**	**346.88**	**303.91**	**342.97**	**303.91**	350.78
	End	538.28	514.84	612.50	514.84	596.88	518.75	**518.75**	518.75	542.19	514.84	600.78	440.63
	Duration	253.91	179.69	300.78	183.59	296.88	175.78	**218.75**	175.78	238.28	171.88	296.88	89.84
DEC	Start	303.91	346.88	307.81	346.88	**303.91**	**342.97**	315.63	331.25	311.72	401.56	311.72	311.72
	End	522.66	456.25	518.75	452.34	530.47	**464.06**	495.31	514.84	**514.84**	491.20	507.03	510.94
	Duration	218.75	109.38	**210.94**	105.47	226.56	**121.09**	179.69	183.59	203.13	89.64	195.31	199.22

relatively stable clustering accuracy and time windows testing on the different intensities of additive noise. Our method, however, is limited in some aspects: the highly overlapped components are challenging to the proposed method as it is for newly developed methods and previously developed approaches. Together, our results show that the proposed method provides a new approach to improve our understanding of the discoverable nature of ERP from noisy data and determine a more reliable time window of ERP.

Another issue with the deep clustering methods is initializing DNN with no ground-truth classification/labeling exists. Commonly, *k*-means [17] is used for initializing and tuning of deep clusterings [2]. However, the random optimized results of the *k*-means-based tuning can affect the learning in the DNNs. A similar issue occurs when initializing the unsupervised deep clustering with a trivial clustering such as *k*-means, the Gaussian mixture model (GMM) [29], and hierarchical clustering [15]. To tackle this issue, we fed the semi-supervised methods and DEC with consensus clustering. The drawback to semi-supervised is that this initialization cannot guarantee to obtain the best labeling results. However, it encourages the network to learn the most powerful features of ERP data. Unsupervised methods can appropriately learn the

Table 6
Standard deviation error (SD) between the estimated time windows by the proposed method, consensus clustering, and the studied deep clustering methods in both ERP data when the different noise strengths are added. The proposed method (Ens_DC) has achieved better stability in estimating the time window, especially in the real data.

Method	Properties (ms)	Simulated data		Real data	
		Cond1	Cond2	Cond1	Cond2
Ens_DC	Start	**2.41**	**0.00**	**3.19**	5.38
	End	**1.20**	**1.95**	**3.27**	**5.88**
	Duration	**3.61**	**1.95**	**3.84**	**1.59**
CC	Start	2.56	0.95	11.23	**1.59**
	End	1.95	2.29	41.02	21.07
	Duration	4.11	2.95	37.91	21.05
MLP_FC	Start	3.10	**0.00**	12.94	4.28
	End	1.76	2.09	36.51	22.47
	Duration	2.95	2.09	36.73	23.27
1D CNN	Start	5.04	**0.00**	9.70	3.84
	End	1.91	2.09	34.11	18.61
	Duration	4.34	2.09	29.00	18.10
LSTM	Start	3.61	**0.00**	8.44	3.19
	End	1.91	2.09	35.26	12.78
	Duration	4.09	2.09	37.22	13.40
AE	Start	3.81	7.33	4.73	4.73
	End	2.41	6.38	9.46	22.91
	Duration	5.90	13.50	13.30	23.13
VAE	Start	3.54	3.43	9.05	7.27
	End	4.02	2.86	39.66	31.00
	Duration	7.18	6.20	35.35	35.94
DEC	Start	6.08	3.19	4.73	29.95
	End	5.67	2.73	12.35	27.80
	Duration	11.35	5.67	16.88	45.31

important features of the data with roughly less accuracy than semi-supervised methods.

From the cognitive process perspective, the proposed method resulted in interpretable and reasonable findings based on prior studies. Notably, the statistical analysis results on the artificial data or real data can be crucial when there is a large effect of ERP component(s) due to obtaining uninterpretable statistical differences (e.g., obtaining extremely significant p-value between the parameters). Although this might be considered a better performance of the new methods, further sophisticated statistical analysis of temporal and sensory parameters can provide more information on the effect of interesting ERP. In the simulated data, the determined time windows from the cluster maps (even noisy conditions) were qualified by our previous findings [31] and the pre-defined components' properties. Additionally, the P3 component was reliably identified in both target and non-target conditions, which is interpretable with the purpose of the experiment and findings from the prior study [19]. Noteworthy, the generated responses from different groups and conditions could differ according to the neurological and experimental mechanisms [4,53]. Therefore, the cluster maps from the same potential can emerge in different temporal and spatial properties, e.g., in the ERP study by Koenig et al. [23]. The reason to study the P3 components is that the results are comparable with the ground truth results (in the simulated data) and interpretable for the real data. Therefore, the proposed method is not limited to identifying P3; it can be applied to identifying some other ERP components, such as N2, P2, and N4, in the simulated data. N4, for example, is identified by maps 3 (for 'Cond1′ and 'Cond2′) in Fig. 4**B** and maps 5 in Fig. 4**D**. Similarly, in real data, the identification of the P1 component by maps 4 in Fig. 5**B** and Fig. 5**D** can be discussed using the proposed method.

Considering the likelihood of obtaining imperfect clustering results, even using state-of-the-art clustering methods (including deep clustering), our method provides a confident result and time window determination for testing the researchers' hypotheses. Our early findings showed that combining different deep clustering methods can be useful for processing ERP data. One important advantage of ensemble learning is that different combinations of clustering methods, including deep clustering, are possible in our pipeline even with an unknown number of clusters. This study provides a positive message about using deep clustering methods for processing ERP data. We have provided a GitHub repository (https://github.com/remahini/Deep-Clustering) for the deep clustering methods used in this study, which can be used as a toolbox by changing the input and initializing parameters to be used by the researchers.

Table 7
Spatial correlation between the mean topography map in the ground truth time windows of P3 and the mean topography maps in the obtained time windows acquired by the proposed method, consensus clustering, and the studied deep clustering methods in the simulated data with different additional noises.

Method	No noise		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5dB	
	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2
Ens_DC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.998	0.999	0.998	0.997	0.982
CC	1.000	1.000	1.000	0.998	0.995	1.000	0.994	0.998	0.994	0.998	0.994	0.982
FC_MLP	0.996	0.999	0.998	0.999	0.994	0.996	0.995	0.998	0.986	0.996	0.998	0.986
1D CNN	0.997	0.999	0.998	0.995	0.995	0.999	0.996	0.998	0.991	0.998	0.996	0.987
LSTM	0.997	0.999	0.995	1.000	0.997	0.999	0.997	0.998	0.990	0.998	0.998	0.987
AE	1.000	0.998	1.000	0.991	1.000	0.997	1.000	0.997	0.999	0.996	0.998	0.979
VAE	1.000	0.996	0.991	0.997	0.995	0.993	1.000	0.993	0.998	0.993	0.995	0.989
DEC	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.997	0.999	0.949	0.998	0.992

Table 8
Spatial correlation of mean topography map in the ground-truth time window and mean topography maps in the identified time windows of the studied methods for P3 in the real data with different additional noises.

Method	No noise		SNR = 20 dB		SNR = 10 dB		SNR = 5 dB		SNR = 0 dB		SNR = -5 dB	
	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2	Cond1	Cond2
Ens_DC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.987	0.995	0.998	0.999
CC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.987	0.995	0.959	0.992
MLP_FC	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	1.000	0.998	0.999
1dCNN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	1.000	0.998	0.999
LSTM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999	1.000	0.998	0.999
AE	0.999	1.000	1.000	1.000	0.999	1.000	1.000	1.000	0.999	1.000	0.997	0.999
VAE	0.999	1.000	0.999	1.000	0.999	1.000	0.999	1.000	0.999	0.999	0.998	0.999
DEC	0.999	0.999	0.999	1.000	0.999	0.999	0.999	1.000	0.998	0.999	0.995	0.999

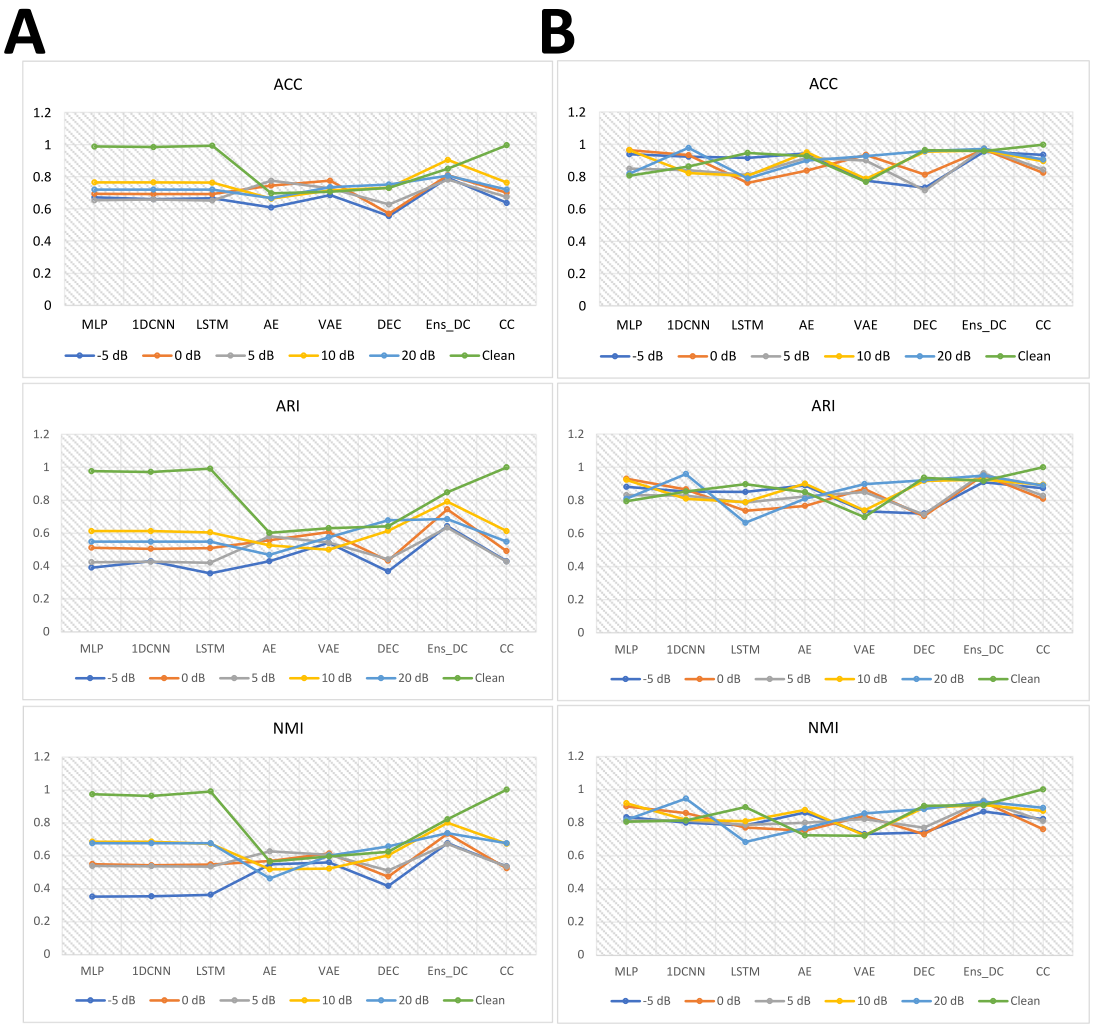

Fig. 6. The performance assessment results, in comparison to the ground truth clustering, for the studied clustering methods from the simulated (left panel) and the real ERP (right panel) data with different additive noise levels. **A.** the accuracy (ACC), adjusted rand index (ARI), and normalized mutual information (NMI) comparison results for clustering results in the simulated data. **B.** the performance metrics (ACC, ARI, and NMI) assessment results for the real data. Noticeably, the proposed clustering provides relatively stable and superior results (except when no noise is added) from both applied data.

Table 9
Illustration of the statistical analysis results of the identifying P3 effect from the measured mean amplitude in the estimated time windows and the Cz and CPz electrode sites from the studied clustering methods on the simulated ERP data at different noise levels.η_p^2 = Partial Eta Squared.

Noise	Method	Ens_DC	CC	FC_MLP	1D CNN	LSTM	AE	VAE	DEC
No noise	F(1,19)	81,317	81,317	199,263	199,263	199,263	235,023	135,063	80,934
	p-value	5.72E-36	5.72E-36	1.15E-39	1.15E-39	1.15E-39	2.39E-40	4.62E-38	5.98E-36
	η_p^2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
SNR = 20 dB	F(1,19)	68,237	68,237	68,237	68,237	68,237	62,941	118,338	75,717
	p-value	3.02E-35	3.02E-35	3.02E-35	3.02E-35	3.02E-35	6.52E-35	1.62E-37	1.13E-35
	η_p^2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
SNR = 10 dB	F(1,19)	27,965	27,965	27,965	27,965	27,965	15,805	15,532	22,446
	p-value	1.44E-31	1.44E-31	1.44E-31	1.44E-31	1.44E-31	3.25E-29	3.83E-29	1.16E-30
	η_p^2	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999
SNR = 5 dB	F(1,19)	16,285	14,151	14,151	12,442	12,442	13,330	9748	10,789
	p-value	2.45E-29	9.27E-29	9.27E-29	3.15E-28	3.15E-28	1.64E-28	3.18E-27	1.22E-27
	η_p^2	0.999	0.999	0.999	0.998	0.998	0.999	0.998	0.998
SNR = 0 dB	F(1,19)	1975	2784	2268	2268	2121	2339	2924	2892
	p-value	1.15E-20	4.50E-22	3.11E-21	3.11E-21	5.86E-21	2.33E-21	2.83E-22	3.15E-22
	η_p^2	0.990	0.993	0.992	0.992	0.991	0.992	0.994	0.993
SNR = -5dB	F(1,19)	883	856	989	1001	989	753	832	608
	p-value	2.15E-17	2.89E-17	7.51E-18	6.73E-18	7.51E-18	9.50E-17	3.74E-17	6.92E-16
	η_p^2	0.979	0.978	0.981	0.981	0.981	0.975	0.978	0.970

Table 10
Illustration of the statistical analysis results and the identified P3 effect measured by mean amplitude in the estimated time windows and Pz electrode from the studied clustering methods for the real ERP data at different noise levels.

Noise	Method	Ens_DC	CC	FC_MLP	1D CNN	LSTM	AE	VAE	DEC
No noise	F(1,39)	121.18	122.01	91.96	93.59	94.21	129.01	121.77	119.80
	p-value	1.57E-13	1.42E-13	8.27E-12	6.49E-12	5.91E-12	6.16E-14	1.46E-13	1.87E-13
	η_p^2	0.76	0.76	0.70	0.71	0.71	0.77	0.76	0.75
SNR = 20 dB	F(1,39)	121.02	127.47	100.69	98.39	92.41	134.86	106.20	120.56
	p-value	1.61E-13	7.38E-14	2.32E-12	3.22E-12	7.73E-12	3.15E-14	1.08E-12	1.70E-13
	η_p^2	0.76	0.77	0.72	0.72	0.70	0.78	0.73	0.76
SNR = 10 dB	F(1,39)	120.12	95.17	103.72	96.16	94.90	128.98	111.35	117.88
	p-value	1.79E-13	5.13E-12	1.52E-12	4.44E-12	5.34E-12	6.18E-14	5.46E-13	2.37E-13
	η_p^2	0.75	0.71	0.73	0.71	0.71	0.77	0.74	0.75
SNR = 5 dB	F(1,39)	121.73	99.19	104.64	98.46	105.05	128.78	130.45	135.42
	p-value	1.47E-13	2.87E-12	1.34E-12	3.19E-12	1.27E-12	6.33E-14	5.21E-14	2.95E-14
	η_p^2	0.76	0.72	0.73	0.72	0.73	0.77	0.77	0.78
SNR = 0 dB	F(1,39)	121.20	98.54	80.31	102.62	93.91	118.44	128.88	122.95
	p-value	1.57E-13	3.15E-12	5.19E-11	1.77E-12	6.18E-12	2.21E-13	6.26E-14	1.27E-13
	η_p^2	0.76	0.72	0.67	0.72	0.71	0.75	0.77	0.76
SNR = -5dB	F(1,39)	124.33	82.42	122.85	114.88	121.12	132.33	88.62	132.81
	p-value	1.07E-13	3.67E-11	1.28E-13	3.46E-13	1.59E-13	4.19E-14	1.38E-11	3.97E-14
	η_p^2	0.76	0.68	0.76	0.75	0.76	0.77	0.69	0.77

5. Conclusions

This research proposed an ensemble deep clustering methodology for qualifying ERP of interest from grand averaged spatio-temporal ERP data. The proposed method has been successfully applied to the simulated ERP and the real ERP data to assess and compare previous findings. Our findings suggested that the time window of ERP can be identified using ensemble deep clustering while a considerable amount of noise exists after preprocessing. Compared to the state-of-the-art clustering methods, the proposed method obtained superior results in terms of the temporal properties of the time windows and clustering performance. The robust clustering performance of the proposed method discloses its confidential properties for use in ERP data. Yet, studying the ensemble deep clustering in the subject, single-trial, and electrode resolution is an open question. Our further outline is to modify the current design to more sophisticated data, e.g., single-trial EEG data.

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.

Acknowledgments

This study would like to remember Prof. Tapani Ristaniemi, who was involved in this study and passed away in 2020, for his great help to all the authors, especially Fengyu Cong, Asoke K. Nandi, Timo Hämäläinen, and Reza Mahini.

References

[1] B. Abu-Jamous, R. Fa, D.J. Roberts, A.K. Nandi, M-N scatter plots technique for evaluating varying-size clusters and setting the parameters of Bi-CoPaM and Uncles methods, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, https://doi.org/10.1109/ICASSP.2014.6854902.
[2] E. Aljalbout, V. Golkov, Y. Siddiqui, D.J.a.p.a. Cremers. Clustering with Deep Learning: Taxonomy and New Methods, 2018. https://doi.org/arXiv:1801.07648v2.
[3] P. Bashivan, I. Rish, M. Yeasin, N. Codella, Learning representations from EEG with deep recurrent-convolutional neural networks, 2015. arXiv preprint arXiv:1511.06448. https://doi.org/10.48550/arXiv.1511.06448.
[4] C. Berchio, A.-L. Küng, S. Kumar, P. Cordera, A.G. Dayer, J.-M. Aubry, C.M. Michel, C. Piguet, Eye-gaze processing in the broader bipolar phenotype revealed by electrical neuroimaging, Psychiat. Res.: Neuroimag. 291 (2019) 42–51, https://doi.org/10.1016/j.pscychresns.2019.07.007.
[5] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).
[6] J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, 1981. https://doi.org/10.1007/978-1-4757-0450-1.
[7] M.A. Boudewyn, S.J. Luck, J.L. Farrens, E.S. Kappenman, How many trials does it take to get a significant ERP effect? It depends. 55(6) (2018) e13049. https://doi.org/10.1111/psyp.13049.
[8] Y. Cao, T.A. Geddes, J.Y.H. Yang, P. Yang, Ensemble deep learning in bioinformatics, Nat. Mach. Intell. 2 (9) (2020) 500–508, https://doi.org/10.1038/s42256-020-0217-y.
[9] H. Cecotti, A. Graser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, IEEE Trans. Patt. Anal. Mach. Intell. 33 (3) (2011) 433–445, https://doi.org/10.1109/TPAMI.2010.125.
[10] A. Delorme, S. Makeig, Mar). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis [Article], J. Neurosci. Methods 134 (1) (2004) 9–21, https://doi.org/10.1016/j.jneumeth.2003.10.009.
[11] M. Dinov, R. Leech, Modeling uncertainties in EEG microstates: analysis of real and imagined motor movements using probabilistic clustering-driven training of probabilistic neural networks, [Methods]. 11 (534) (2017), https://doi.org/10.3389/fnhum.2017.00534.
[12] K.G. Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Computer Vision (ICCV), 2017 IEEE International Conference on, 2017, https://doi.org/10.1109/ICCV.2017.612.
[13] V. Duddu, N. Rajesh Pillai, D.V. Rao, V.E. Balas, Fault tolerance of neural networks in adversarial settings, J. Intell Fuzzy Syst 38 (5) (2020) 5897–5907, https://doi.org/10.3233/jifs-179677.
[14] A.B. Geva, H. Pratt, Unsupervised clustering of evoked potentials by waveform, 1994, Med Biol. Eng. Comput. 32 (5) (1994) 543–550, https://doi.org/10.1007/BF02515313.
[15] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, J. Yin, Adaptive self-paced deep clustering with data augmentation, IEEE Trans. Knowl. Data Eng. 1–1 (2019), https://doi.org/10.1109/tkde.2019.2911833.
[16] M.K. Islam, A. Rastegarnia, Z. Yang, Methods for artifact detection and removal from scalp EEG: a review, 2016/11/01/), Neurophysiol. Clinique/Clinical Neurophysiol. 46 (4) (2016) 287–305, https://doi.org/10.1016/j.neucli.2016.07.002.
[17] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666, https://doi.org/10.1016/j.patrec.2009.09.011.
[18] R.E. Kallionpää, H. Pesonen, A. Scheinin, N. Sandman, R. Laitio, H. Scheinin, A. Revonsuo, K. Valli, Single-subject analysis of N400 event-related potential component with five different methods, Int. J. Psychophysiol. 144 (2019) 14–24, https://doi.org/10.1016/j.ijpsycho.2019.06.012.
[19] E.S. Kappenman, J.L. Farrens, W. Zhang, A.X. Stewart, S.J. Luck, ERP CORE: an open resource for human event-related potential research, 2021/01/15/), Neuroimage 225 (2021), 117465, https://doi.org/10.1016/j.neuroimage.2020.117465.
[20] E.S. Kappenman, S.J. Luck, ERP components: The ups and downs of brainwave recordings (2012) 3-30. https://doi.org/10.1093/oxfordhb/9780195374148.013.0014.
[21] A. Kiesel, J. Miller, P. Jolicœur, B. Brisson, Measurement of ERP latency differences: a comparison of single-participant and jackknife-based scoring

methods. 45(2) (2008) 250-274. https://doi.org/10.1111/j.1469-8986.2007.00618.x.
[22] D. Kingma, M. Welling. Auto-encoding variational bayes, ArXiv: 13126114. The 2nd International Conference on Learning Representations, 2013. https://doi.org/10.48550/arXiv.1312.6114.
[23] T. Koenig, M. Stein, M. Grieder, M. Kottlow, A tutorial on data-driven methods for statistically assessing ERP topographies, Brain Topography 27 (1) (2014) 72–83, https://doi.org/10.1007/s10548-013-0310-1.
[24] T. Kohonen, THE SELF-ORGANIZING MAP, Proc IEEE 78 (9) (1990) 1464–1480, https://doi.org/10.1109/5.58325.
[25] H.W.J.N.r.l.q. Kuhn, The Hungarian method for the assignment problem. *2*(1-2) (1955) 83-97.
[26] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. https://www.nature.com/articles/nature14539.pdf.
[27] D. Lehmann, Brain Electric Microstates and Cognition: The Atoms of Thought. In: E. R. John, T. Harmony, L. S. Prichep, M. Valdés-Sosa, & P. A. Valdés-Sosa (Eds.), Machinery of the Mind: Data, Theory, and Speculations About Higher Brain Function (pp. 209-224). Birkhäuser Boston, 1990. https://doi.org/10.1007/978-1-4757-1083-0_10.
[28] F. Li, R. Yan, R. Mahini, L. Wei, Z. Wang, K. Mathiak, R. Liu, F. Cong, End-to-end sleep staging using convolutional neural network in raw single-channel EEG, Biomed. Signal Process. Control 63 (2021), 102203, https://doi.org/10.1016/j.bspc.2020.102203.
[29] K.L. Lim, X. Jiang, C. Yi, Deep clustering with variational autoencoder, IEEE Signal Process. Lett. 27 (2020) 231–235, https://doi.org/10.1109/LSP.2020.2965328.
[30] S.J. Luck, An introduction to the event-related potential technique, (Second edition ed.)., MIT press. (MIT press), 2014.
[31] R. Mahini, Y. Li, W. Ding, R. Fu, T. Ristaniemi, A.K. Nandi, G. Chen, F. Cong, Determination of the time window of event-related potential using multiple-set consensus clustering [Methods], 2020-October-21, Front. Neurosci. 14 (1047) (2020), https://doi.org/10.3389/fnins.2020.521595.
[32] R. Mahini, P. Xu, G. Chen, Y. Li, W. Ding, L. Zhang, N.K. Qureshi, T. Hämäläinen, A.K. Nandi, F. Cong, Correction: optimal number of clusters by measuring similarity among topographies for spatio-temporal ERP analysis, 2022/11/01), Brain Topography 35 (5) (2022) 558, https://doi.org/10.1007/s10548-022-00918-9.
[33] R. Mahini, P. Xu, G. Chen, Y. Li, W. Ding, L. Zhang, N.K. Qureshi, T. Hämäläinen, A.K. Nandi, F. Cong, Optimal number of clusters by measuring similarity among topographies for spatio-temporal ERP analysis. Brain Topography (2022b). https://doi.org/10.1007/s10548-022-00903-2.
[34] S. Makeig, A. Bell, T.-P. Jung, T.J. Sejnowski, Independent component analysis of electroencephalographic data, Adv. Neural Inform. Process. Syst. 8 (1995).
[35] P. Masulli, F. Masulli, S. Rovetta, A. Lintas, A.E.P. Villa, Fuzzy clustering for exploratory analysis of EEG event-related potentials, IEEE Trans. Fuzzy Syst. 28 (1) (2020) 28–38, https://doi.org/10.1109/TFUZZ.2019.2910499.
[36] M.C. Medeiros, M. McAleer, D. Slottje, V. Ramos, J. Rey-Maquieira, An alternative approach to estimating demand: neural network regression with conditional volatility for high frequency air passenger arrivals, 2008, J. Economet. 147 (2) (2008) 372–383, https://doi.org/10.1016/j.jeconom.2008.09.018.
[37] M. Meila, Comparing clusterings – an information based distance, J. Multivariate Anal. 98 (5) (2007) 873–895, https://doi.org/10.1016/j.jmva.2006.11.013.
[38] C.M. Michel, T. Koenig, EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: a review, Neuroimage 180 (2018) 577–593, https://doi.org/10.1016/j.neuroimage.2017.11.062.
[39] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, J.J.I.A. Long, A survey of clustering with deep learning: from the perspective of network architecture 6 (2018) 39501-39514. https://doi.org/10.1109/ACCESS.2018.2855437.
[40] N. Mrabah, N.M. Khan, R. Ksantini, Z. Lachiri, Deep clustering with a dynamic autoencoder: from reconstruction towards centroids construction, Neural Networks 130 (2020) 206–228, https://doi.org/10.1016/j.neunet.2020.07.005.
[41] Y. Mu, S. Han, Neural oscillations involved in self-referential processing, Neuroimage 53 (2) (2010) 757–768, https://doi.org/10.1016/j.neuroimage.2010.07.008.
[42] M.M. Murray, D. Brunet, C.M. Michel, Topographic ERP analyses: a step-by-step tutorial review, Brain Topography 20 (4) (2008) 249–264, https://doi.org/10.1007/s10548-008-0054-5.
[43] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems, 2002.
[44] G. Oetken, T. Parks, H. Schussler, New results in the design of digital interpolators, IEEE Trans. Acoust. Speech Signal Process. 23 (3) (1975) 301–309, https://doi.org/10.1109/tassp.1975.1162686.
[45] R. Oostenveld, P. Fries, E. Maris, J.-M. Schoffelen, FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data, Comput. Intell. Neurosci. 156869 (2011). https://doi.org/10.1155/2011/156869.
[46] R.D. Pascual-Marqui, C.M. Michel, D. Lehmann, Segmentation of brain electrical activity into microstates: model estimation and validation, IEEE Trans. Biomed. Eng. 42 (7) (1995) 658–665, https://doi.org/10.1109/10.391164.
[47] R.D. Pascual-Marqui, C.M. Michel, D.J.I.T.o.B.E. Lehmann, Segmentation of brain electrical activity into microstates: model estimation and validation. 42(7) (1995) 658-665. https://doi.org/10.1109/10.391164.
[48] J.M. Pena, J.A. Lozano, P.J.P.r.l. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm. 20(10) (1999) 1027-1040. https://doi.org/10.1016/S0167-8655(99)00069-0.
[49] S.M. Peterson, R.P. Rao, B.W. Brunton, Learning neural decoders without labels using multiple data streams, J. Neural Eng. 19(4) (2022) 046032. https://doi.org/DOI 10.1088/1741-2552/ac857c.
[50] Y. Qi, F. Luo, W. Zhang, Y. Wang, J. Chang, D.J. Woodward, A.C. Chen, J. Han, Sliding-window technique for the analysis of cerebral evoked potentials, Beijing Da Xue Xue Bao Yi Xue Ban 35 (3) (2003) 231–235.
[51] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P.S. Yu, L. He, Deep clustering: a comprehensive survey, 2022. arXiv preprint arXiv:2210.04142. https://doi.org/10.48550/arXiv.2210.04142.
[52] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T.H. Falk, J. Faubert, Deep learning-based electroencephalography analysis: a systematic review, J. Neural Eng.. (2019). https://orcid.org/0000-0003-4408-5221.
[53] P. Ruggeri, H.B. Meziane, T. Koenig, C. Brandner, A fine-grained time course investigation of brain dynamics during conflict monitoring, Article 3667, Scientific Reports 9 (2019), https://doi.org/10.1038/s41598-019-40277-3.
[54] O. Sagi, L. Rokach, Ensemble learning: a survey, WIREs Data Min. Knowl. Disc. 8 (4) (2018) e1249.
[55] S.B. Shaw, K. Dhindsa, J.P. Reilly, S. Becker, Capturing the forest but missing the trees: microstates inadequate for characterizing shorter-scale EEG dynamics, Neural Computat. 31 (11) (2019) 2177–2211, https://doi.org/10.1162/neco_a_01229.
[56] A. Sikka, H. Jamalabadi, M. Krylova, S. Alizadeh, J.N. van der Meer, L. Danyeli, M. Deliano, P. Vicheva, T. Hahn, T. Koenig, D.R. Bathula, M. Walter, Investigating the temporal dynamics of electroencephalogram (EEG) microstates using recurrent neural networks, Human Brain Mapping (2020), https://doi.org/10.1002/hbm.24949.
[57] A. Strehl, J. Ghosh, Cluster ensembles- a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3(3) (2003) 583-617. https://doi.org/10.1162/153244303321897735.
[58] L. Tan, J. Jiang, Chapter 11 - Multirate Digital Signal Processing, Oversampling of Analog-to-Digital Conversion, and Undersampling of Bandpass Signals, in: L. Tan, J. Jiang (Eds.), Digital Signal Processing, Third Edition, Academic Press, 2019, pp. 529–590, https://doi.org/10.1016/B978-0-12-815071-9.00011-7.
[59] M.C. Thomas, A.T. Joy. Elements of information theory, Wiley-Interscience, 2006.
[60] R. Tibshirani, G. Walther, Cluster validation by prediction strength, 2005/09/01, J. Computat. Graph. Statist. 14 (3) (2005) 511–528, https://doi.org/10.1198/106186005X59243.
[61] A. Topchy, A.K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1866–1881, https://doi.org/10.1109/TPAMI.2005.237.
[62] A. Tzovara, M.M. Murray, C.M. Michel, M. De Lucia, A tutorial review of electrical neuroimaging from group-average to single-trial event-related potentials, 2012/08/01, Dev. Neuropsychol. 37 (6) (2012) 518–544, https://doi.org/10.1080/87565641.2011.636851.
[63] N.X. Vinh, J. Epps, J.J.T.J.o.M.L.R. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. 11 (2010) 2837-2854.
[64] A.J. Wills, A. Lavric, Y. Hemmings, E. Surrey, Attention, predictive learning, and the inverse base-rate effect: evidence from event-related potentials, Neuroimage 87 (2014) 61–71, https://doi.org/10.1016/j.neuroimage.2013.10.060.
[65] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis. International Conference on Machine Learning, 2016.
[66] D. Yao, Y. Qin, S. Hu, L. Dong, M.L. Bringas Vega, P.A. Valdés Sosa, Which reference should we use for EEG and ERP practice?, 2019/07/01, Brain Topography 32 (4) (2019) 530–549, https://doi.org/10.1007/s10548-019-00707-x.
[67] P. Zhang, X. Wang, W. Zhang, J. Chen, Learning spatial–spectral–temporal EEG features with recurrent 3D convolutional neural networks for cross-task mental workload assessment, IEEE Trans. Neural Syst. Rehabilitat. Eng. 27 (1) (2019) 31–42, https://doi.org/10.1109/TNSRE.2018.2884641.
[68] A. Khanna, A. Pascual-Leone, F. Farzan, Reliability of Resting-State Microstate Features in Electroencephalography, PloS One 9 (12) (2014) e114163, https://doi.org/10.1371/journal.pone.0114163.