frontiers

Supplementary Material

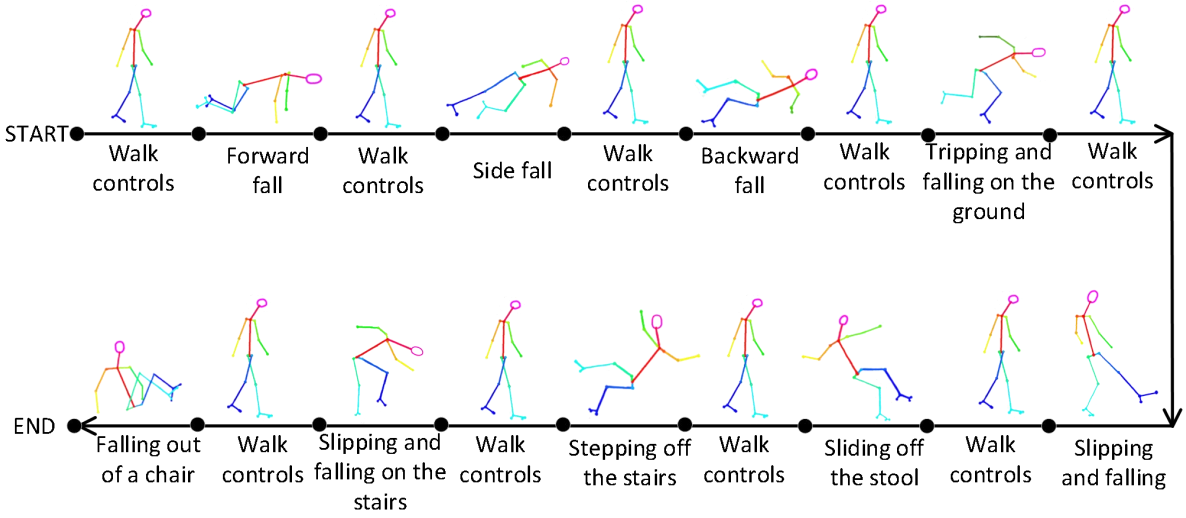
Patterns of neural activity associated with anticipated falls

Haiyan Yu[1]*, Xingyuan Deng[1], Jinxiang Tang[2], Yixiao Fu[3], Yuxiang Cai[1], Ruofan Wang[4], Faguo Yue[2]*

*** Correspondence:** Haiyan Yu: yhy188@gmail.com (H.Y. Yu), Faguo Yue:mnyfg@163.com(F. G., Yue)

1 Design table of anticipated-falls in the experiments

In the design of the experiment, each trial consists of nine pairs of anticipated falls and normal walks (Supplementary Figure 1.). The informed written consent from all recruiters was obtained prior to the research with the assigned project (A2021-287). This experiment is approval from our institutional ethics committee under Dr. Zhangyong Li.



Supplementary Figure 1. Nine pairs of anticipated falls and normal controls in the experimental design.

2 Artifact removal with second-order blind identification

Let $s(t)$ be the output of the blind source separation (BSS) process with input $x(t)$. For each wave, the energy in the time domain is computed by $E = \int_{-\infty}^{+\infty} s^2(t)dt$, where s(t) is the element of the sub-waveform of EEG g[t] in the time domain after the blind source separation.

The second-order blind identification (SOBI) algorithm is a process of diagonalizing a delay correlation matrix with delay time t. By selecting several different delay times t, the corresponding

delay correlation matrices are simultaneously joint diagonalized, making identical diagonal elements non-existent. The procedures of the SOBI algorithm are as follows:

(1) Obtaining the Whitening matrix W. Let $z(t) = Wx(t)$, where W is the whitening matrix. The covariance matrix of $z(t)$ is a unit matrix that has the second-order correlation between components removed. Thus, the matrix can be obtained by the reverse process of $z(t) = Wx(t)$.

(2) Computing the sampling covariance matrix. For a fixed delay $\mu = \{\mu_l | l = 1,2,\dots,L\}$, the algorithm obtains the sampling covariance matrix of the whitened data by

$$R(\mu) = E[z(t + \mu)z^T(t)] = ARz(\mu)A^T$$

where A is the mixture matrix.

(3) Computing the orthogonal matrix U. For all $R(\mu)$, the joint approximate diagonalization algorithm is employed to obtain matrix U, $U^T R(\mu_l)U = D_l$, where $\{ D_l\}$ is a set of diagonal matrices.

(4) Computing the mixture matrix A. Since $z(t) = U^T Wx(t)$, $A = W^+ U$, where W^+ is the pseudo-inverse matrix of the separation matrix.

After obtaining the relevant source signal $s(t)$, the algorithm removes the redundant independent components of the source signal. Subsequently, the algorithm reconstructs the data with $x_r(t) = W^+ s(t)$, where $x_r(t)$ is the reconstructed observation vector and $s(t)$ is the new independent source matrix after setting all the redundant components of the source signal with zeroes.

3 Feature acquisition of EEG with entropy

Let F be the feature set of the feature acquisition with input $s[t]$, $t = 1,\dots,T$. Here three types of entropy of the data after BSS are employed as the features, $F(s[t]) = \{PeEn, ApEn, SaEn\}$, which represents the permutation entropy (PeEn), approximate entropy (ApEn) and sample entropy (SaEn).

Permutation entropy is a sequential analysis method that describes the regularity of signals according to local temporal data. Let $P = \{p(\pi_j)\}$ be the probability distribution whose frequencies are related to the possible permutation patterns (Henry and Judge, 2019). Permutation entropy can be expressed as:

$$PeEn(s[t], m, \pi) = -\sum_{j=1}^{m!} p(\pi_j) \cdot lnp(\pi_j)$$

where π is the embedded time delay that determines the time separation between $s[t]$ values, $1 \leq j \leq m!$, m is the embedding dimension that controls the length of each of the new column vectors ($m > 1$), and $ln(\cdot)$ is the logarithm function. Here $m = 3$ is a recommended value. The smaller the permutation entropy value, the more regular the sequence. The probability can be expressed as

$$p(\pi_j) = \frac{\#\{q | qT - (m-1);\ (q) has\ type\ \pi_j\}}{T - (m-1)}$$

where the symbol $\#\{\cdot\}$ stands for the number of elements in the set, and (q) is a set of ordinal patterns of order m and is generated by (q): $(s_{q-(m-1)}, s_{q-(m-2)}, \ldots, s_{q-1}, s_q)$. This set is the m-dimensional vector of values at times $s, s-1, \ldots, s-(m-1)$.

Furthermore, approximate entropy (ApEn) is employed to quantify the complexity and irregularity of the EEG temporal data. The larger the approximate entropy, the more complex the sequence, the more irregular the signal fluctuations, and the more active the EEG. Given the data $s[t](1 \le t \le T)$, let $s_m(t) = \{s(t), s(t+1), \ldots, s(t+m-1)\}(t = 1,2,\ldots,T-m+1)$ be its embedded m-dimensional vector. Thus, the approximate entropy can be expressed as:

$$ApEn(m, r, T) = \Phi^m(r) - \Phi^{m+1}(r)$$

where m is the embedding dimension, r is the threshold parameter, T is the length of the temporal data,

$$\begin{cases} \Phi^m(r) = \dfrac{\sum_{t=1}^{T-m+1} ln\, C_t^m(r)}{T-m+1} \\ \Phi^{m+1}(r) = \dfrac{\sum_{t=1}^{T-m} ln\, C_t^{m+1}(r)}{T-m} \end{cases}$$

$C_t^m(r)$ is the probability of which the distance between vectors $s_m(t)$ and $s_m(v)$ is less than the given threshold r,

$$C_t^m(r) = \frac{\sum_{v=1}^{T-m+1} \theta(d_{tv}^m - r)}{T-m+1}$$

and $\theta(d_{tv}^m - r) = \begin{cases} 1, (d_{tv}^m - r) \ge 0, \\ 0, (d_{tv}^m - r) < 0. \end{cases}$

Given $k = 0,1,\ldots,m$, the distance between the two vectors $s_m(t)$ and $s_m(v)$ is

$d_{tv}^m = d[s_t^m, s_v^m] = max\,(\,|s(t+k) - s(v-k)|)$.

Here the values of the parameters are selected as $m = 2, r = 0.2SD$ (where SD is the standard deviation of the temporal data).

Sample entropy is an adapted version of approximate entropy, which is also used to measure the complexity of temporal data. The lower the sample entropy, the more complex the temporal data. Given the data $s[t](1 \le t \le T)$, and its embedded m-dimensional vector $s_m(t)$, the sample entropy can be expressed as:

$$SaEn(m, r, T) = -\, ln\left[\frac{B^{m+1}(r)}{B^m(r)}\right]$$

where m is the embedding dimension, r is the threshold parameter, and $B^m(r)$ is the total number of matching templates.

$$\begin{cases} B^m(r) = \dfrac{\sum_{t=1}^{T-m+1} C_t^m(r)}{T-m+1} \\ B^{m+1}(r) = \dfrac{\sum_{t=1}^{T-m} C_t^{m+1}(r)}{T-m} \end{cases}$$

where $C_t^m(r)$ is the probability of which the distance between vectors $s_m(t)$ and $s_m(v)$ is less than the given threshold r.

These results indicate superimposing and averaging the ratio of the number of all arbitrary vectors (whose distance is less than r) to the total number of vectors. Similarly, here $m = 2, r = 0.2SD$ for the temporal data.

4 Association of the brain activity with anticipated falls using classifiers

Given a sample *O* with feature set F drawn from a distribution, a loss function is set to find the function $\varphi(\cdot)$ that is a classifier model (from sample *O* projected on features *F*). Thus, the objective is to minimize the risk of prediction, i.e., the mean square prediction error

$$R(O, \varphi) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where the dataset $O = \{(x_i, y_i)|i = 1, \ldots, N\}$, x_i and y_i are the predictors and outcomes (e.g., anticipated falls or walks), $\hat{y}_i = \varphi(\cdot)$ and $\varphi(x)$ is the classifier. Using the acquired features of entropy, the classifier can be represented as $\hat{y}_l = \widehat{\varphi}(x_l) = \varphi(\{PeEn, ApEn, SaEn\})$. The classifier can be selected from the state-of-the-art machine learning algorithms, such as random forests (RF), support vector machines (SVM), and k-nearest neighbors (k-NN).

Alternatives measures of evaluating ML predictions are variants of the confusion matrix. For example, the classification accuracy can be expressed as $Acc(O, \varphi) = 1/N \sum_{i=1}^{N} l_{(y_i=\hat{y}_i)}$, where $l_{(y_i=\hat{y}_i)}$ is an indicator function equaling 1 for $y_i = \hat{y}_i$; otherwise 0. This measure (of the possible accuracy) is often presented as a percentage between 0% (for the worst) and 100% (for the best classifiers). Here we explore the random forest in the supervised ML framework with the features of the EEG data. The procedure list is as follows.

(1) Random sampling with Bootstrapping. The algorithm first draws training sets from the original sample set O. In each run, this algorithm uses the Bootstrapping method to obtain *n* training samples from O (sampling with replacement). Thus, we repeatedly obtain R independent training sets, $O_\gamma \in \{O_1, \ldots, O_R\}$.

(2) Training decision tree models. With O_γ, the model $G_\gamma(F)$ is trained with the Gini coefficient of features *F*. Gini coefficients are used as the criterion for providing the best branch split.

$$Gini(p_\gamma) = \sum_{\gamma=1}^{2} p_\gamma (1 - p_\gamma) = 1 - \sum_{\gamma=1}^{2} p_\gamma^{\ 2}$$

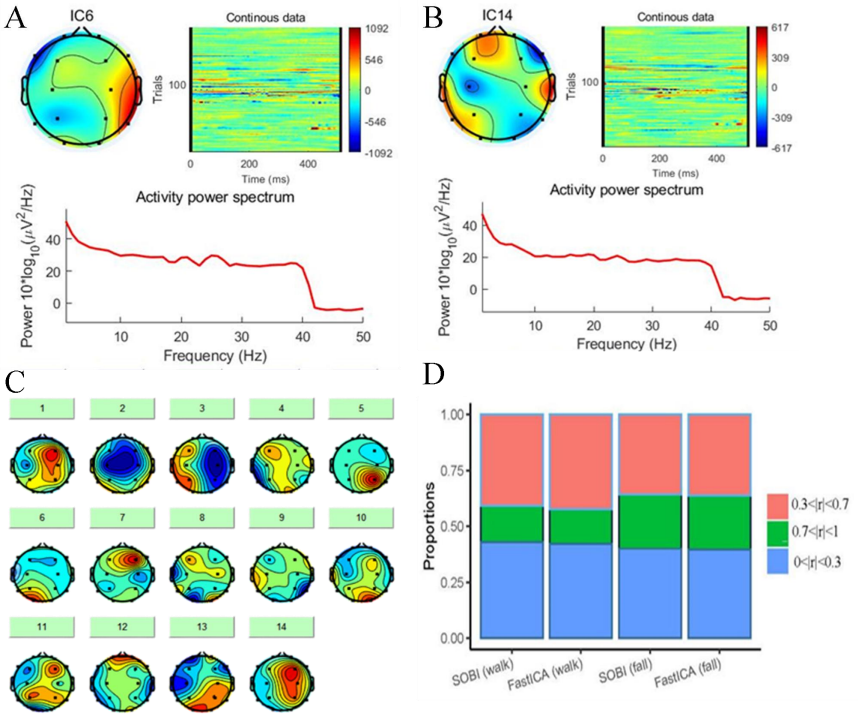
where p_γ represents the proportion of observations in a region from the γ -th class (group), $\gamma \in \{1,2\}$ for binary classifications. The lower the Gini index, the less misclassification the algorithm predicts. Therefore, the training process is to obtain random forests with lower values of the Gini for better predictions.

(3) Aggregating the prediction. With the results of the R decision tree models, the mean of the outputs is obtained by averaging all the prediction results.

The outcomes of these ML algorithms can be further used in the alternative measures of evaluating ML predictions. They can be used to obtain the parameters of the models in training and compare the results in testing.

5 Comparison of ICA analysis for artifact removal

To compare the results of the SOBI algorithm to related methods, the FastICA algorithm was performed on the experimental dataset (Fig. A1). Both methods were employed to remove interference components such as oculoelectric artifacts. The criterion for comparison is Pearson correlation coefficients between the EEG signals extracted by the BSS filters (e.g., SOBI or FastICA) and without them for the anticipated falls and walk controls. The greater the abstract value (|r|) of the Pearson correlation coefficient, the higher the degree of correlation between the two signals. In practice, the proportions of the correlation coefficients are divided into three intervals. Thus, the larger the values in (0.7, 1], the better the BSS algorithm.



Supplementary Figure 2. ICA analysis with FastICA and comparison to SOBI on the EEG data. **(A)** ICA analysis of IC6 with FastICA. **(B)** ICA analysis of IC14 with FastICA. **(C)** ICA analysis with FastICA after removing artifacts. **(D)** Comparison of FastICA to SOBI.

The results in Fig. A1d show that the results with the SOBI algorithm have a higher correlation than those with FastICA. The value of |r| in (0.7, 1] is 0.1617 for SOBI, which is 4.32% higher than FastICA on the correlation between the EEG signals extracted by the BSS filters (e.g., SOBI or FastICA) and without them for the walk controls. Similarly, the value of |r| in (0.7, 1] is 0.2439 for SOBI, which is 1.85% higher than FastICA on the correlation for the anticipated falls. Thus, the proposed method chooses the SOBI algorithm to preprocess the EEG data for further analysis.