Neural Comput & Applic
DOI 10.1007/s00521-016-2767-9

ORIGINAL ARTICLE

Classification of motor imagery electroencephalography signals using spiking neurons with different input encoding strategies

Ruben I. Carino-Escobar[1,2] · Jessica Cantillo-Negrete[1] · Josefina Gutierrez-Martinez[1] · Roberto A. Vazquez[2]

Received: 5 May 2016 / Accepted: 25 November 2016

Abstract Motor imagery-based brain–computer interfaces decode users' intentions from the electroencephalogram; however, poor spatial resolution makes automatic recognition of these intentions a challenging task. New classification approaches with low computational costs and high classification performances need to be developed in order to increase the number of users benefitted by these systems. On the other hand, spiking neuron models, which are mathematical abstractions of real neurons, have shown good performances in several classification tasks, making these models suitable for motor imagery classification. In this work, two different encoding strategies for spiking neuron models, applied to the classification of motor imagery time–frequency features of stroke patients and healthy subjects, were evaluated. Classification performances and computational costs of spiking neuron models were compared against those of linear discriminant analysis, support vector machines and artificial neural networks. Results showed that a time-varying encoding strategy is more suitable for motor imagery classification, and its implementation computational cost is low. Therefore, a spiking neuron model with a time-varying encoding strategy could increase the number of potential users of brain–computer interfaces.

Keywords Brain–computer Interfaces · Wavelet · Particle swarm optimization · Pattern classification

1 Introduction

Brain–computer interfaces (BCI) are systems capable of decoding user's intentions from different regions of the brain in order to control external devices such as wheelchairs or prosthesis. BCI systems are structured by the following stages: brain signal acquisition, preprocessing, feature extraction, classification and external device communication. Stroke rehabilitation is one of the most promising applications for BCI systems. Stroke is the first worldwide cause of disability; its most incapacitating sequel, hemiparesis, is suitable for treatment with BCIs [1]. Most of the proposed BCI systems for stroke rehabilitation are based on electroencephalography (EEG), since it is not invasive and has a good time resolution compared to other methods. However, its poor spatial resolution makes automatic recognition of user's intentions a challenging task.

Motor imagery (MI) is a strategy that BCI users can use in order to encode their intentions in the EEG. This strategy involves the movement intention of an extremity, like a hand or foot, which generates cortical activations similar to those elicited during real movement [2–4]. It is well known that MI produces a decrease (event-related desynchronization) or/and an increase (event-related synchronization) in power in the frequency band from 8 to 30 Hz (mu and beta frequency bands), and it is elicited approximately 500 ms after the beginning of the task [5]. MI is linked to the sensory-motor cortex area of the human brain which can be recorded by central, parietal and temporal electrodes of the international 10–20 electrode position system. Stroke patients can be benefited from a MI-based BCI since it could increase brain plasticity. In order to decode MI from the EEG, extraction of the most discriminative features is performed, followed by classification of such features into

✉ Ruben I. Carino-Escobar
ricarino@inr.gob.mx

[1] Technological Research Subdirection, Instituto Nacional de Rehabilitación, 14389 Mexico City, Mexico

[2] Intelligent Systems Group, Faculty of Engineering, La Salle University, 06140 Mexico City, Mexico

Published online: 16 December 2016

different MI tasks. For feature extraction, methods such as common spatial patterns, Fourier transform and wavelet transform (WT) are used. Particularly, time–frequency information is often extracted by means of WT due to its ability to achieve high resolutions in the time domain while also preserving resolution in the frequency domain of acquired EEG signals [6–9].

For MI classification different approaches have been used, including linear discriminant analysis (LDA) [10], support vector machines (SVM) [11] and artificial neural networks (ANN) [12]. However, the performance of a new type of artificial neuron, the spiking neuron (SN) model, has not been widely tested for MI classification. SN models are the third generation of artificial neurons; these mathematical models abstract the biological behavior of real neurons in a much precise manner than previous generations of artificial neurons, such as the perceptron and radial basis function neurons. The main difference between past generations of artificial neurons and SNs is that a simulated time window is added into the neuron model, and in this way, SNs integrate time into their abstraction models. Input data are introduced to a SN as a simulated electrical current which magnitude may change or may be constant with respect to the simulation time window. The simulation current is afterward included into the SN activation function which generates an output made of a spike train, like a real biological neuron. Pattern recognition can be performed by associating the number of spikes in the output train (rate coding), the time related to those spikes (time coding) or a combination of both, to the simulated time window of the SN model. Nowadays, the SN model with the most realistic behavior is the one proposed by Hodgkin-Huxley; however, its computational cost is the highest among SN models [13]. The SN model with the lowest computational cost is the Integrate & Fire model, but its biological behavior is limited [14]. The Izhikevich model (IZ) has low computational cost and high degree of biological realism, since it is able to accurately simulate the behavior of practically all known biological neurons [15]. Different methodologies have been used to train SNs and networks of SN (SNN), which can be arranged in three groups: (1) SNN trained with a variant of the gradient-descent algorithm [16–19], (2) those trained with the biological inspired spike-time-dependent plasticity (STDP) strategy [20–22] and (3) those trained by evolutionary algorithms, like genetic algorithms, particle swarm optimization (PSO) and the artificial bee colony algorithm [23–27].

Although several authors propose SNs and SNN for classification purposes, most of them have only been tested on benchmark datasets [16, 17, 21, 23–27]. Kasabov et al. described the performance of a SNN architecture for classification of cognitive tasks in the EEG, called the NeuCube. Data were extracted from a sample of 7 healthy subjects, with the NeuCube showing a significant increase in classification performance when compared to other classification methods; however, no MI tasks or information from stroke patients was tested [28]. Chen et al. also used the NeuCube architecture for a pilot study involving classification of a healthy subject wrist MI. They achieved good classification performances; however, stroke patientś MI data were not classified with the NeuCube, nor was the classification performance compared with other classification methods [29]. Therefore, to further assess SN potential for improving EEG classification in neurorehabilitation BCI systems, an evaluation using MI data acquired from stroke patients must be done. Another factor to evaluate, regarding SN implementation in BCI systems, is their computational cost since a BCI system's requirements will be directly affected by it. A lower computational cost for a BCI implementation would decrease its hardware requirements, decreasing therapy costs with this technology to which more potential users could have access.

It is well known that relevant MI features are coded in time, spatial and frequency domains of the EEG; therefore, if time–frequency information is preserved, these features could increase the possibilities of a correct classification of different MI tasks. SN models can preserve this time–frequency information, since it is not necessary to segment it, instead it can be incorporated into the model as a simulation input current, varying within the simulation time window of the model. To the authors' knowledge, this approach has not been used for MI classification of both healthy and stroke patients MI data.

In the present work, MI classification from the EEG of a sample of healthy subjects and stroke patients is presented. Classification was performed with two different models: (1) a SN model for which input data are introduced as a time-varying simulation current with regard to the simulation time window and (2) a SN model for which input data are introduced as a time-unvarying simulation current. This comparison is important since each of the encoding strategies could show different performances for the classification of MI features and could potentially increase the performance of a BCI designed for stroke rehabilitation. Performances for both SN models were compared with other well-established methods such as SVM, LDA and a second generation artificial neural network (ANN).

2 Methods

2.1 EEG data acquisition

EEG data were recorded from six patients with stroke diagnosis ($M = 55.8$, $\mathrm{SD} = 12.6$) and six healthy subjects ($M = 29.3$, $\mathrm{SD} = 1.9$) without antecedents of neurological diseases. All stroke patients had paralysis of one hand. EEG recording was done prior written informed consent by the subjects and approval of the research protocol by the ethics committee of the National Institute of Rehabilitation, Mexico City. A 24-bit resolution biosignal amplifier, model g.USBamp from g.tec, was used. The sampling frequency for all recordings was of 256 Hz. Eleven electrodes were placed over the sensory-motor cortex according to the international 10–20 system (electrode positions correspond to T3, P3, C3, Cz, C4, P4, T4, F3, Fz, F4 and Pz), with a ground placed in the AFz position and the reference electrode placed on the right ear lobe. In order to detect eye artifacts, electrodes were also placed in the orbiculus oculi muscle on both eyes. To assess that no real movements were performed during MI tasks, EMG activity was recorded with electrodes placed above the deep flexor and superficial muscles on fingers of both arms.

2.2 Experimental paradigm

A total of 120 trials of right hand MI (RMI) and 120 trials of left hand MI (LMI) were recorded for each subject. To achieve this, each subject was asked to sit on a comfortable armchair with a computer screen placed 1.50 m in front of them. Each trial lasted 8 s, it initiated with a white cross showed on the computer screen to the subject, after 2 s a short auditory warning tone was listened. In this period of the trial, the subjects were instructed to rest with eyes opened and looking at the cross. In the third second, the cross was replaced by an arrow pointing at either the right or the left direction, which indicated the subject to perform right hand MI or left hand MI, respectively. Arrow orientation was randomly selected and lasted 1.5 s on the screen; it disappeared leaving a black background for 3.5 s while the subjects continued performing MI. After the eighth second of the trial, a blue screen appeared indicating that the subject could relax and blink eyes which lasted for a random interval between 3 and 5 s to prevent habituation. The procedure was based on the Graz paradigm [30].

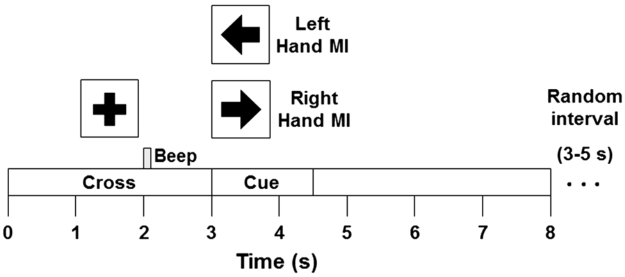

Fig. 1 Timing of the experimental paradigm

Figure 1 shows the timing diagram of the experimental task. All subjects, both stroke patients and healthy subjects, were naïve to MI execution. Healthy subject's data were collected in 2 sessions in which 60 trials of RMI, and LMI was recorded per session, with the two sessions performed within a week. The stroke patient's data were collected in 4 sessions, due to the fact that stroke patients are easily fatigued, for whom 30 trials of RMI and 30 trials of LMI were recorded per session, with all 4 sessions done within a time interval of 2 weeks.

2.3 Signal preprocessing

All EEG signals were re-referenced using the Common Average Reference (CAR) method. CAR consists of subtracting the average of the complete electrode montage from each channel of interest. CAR is capable of producing EEG information that is almost reference-free [31]. After CAR, all channels were conditioned with a 30 Hz low-pass filter, an 8 Hz high-pass filter and two band-stop filters one in the band of 59–61 Hz and another one in the 119–121 Hz frequency band, with all filters being 20th order FIR type.

2.4 Feature extraction

For each trial and channel, the EEG signal was convoluted by complex Morlet wavelets $w(t,f_0)$ having a Gaussian shape in both the time (σ_t) and the frequency domain ($1/2\pi\sigma_t$) around its central frequency $f : w(t,f_0) = A\left[\exp\left(-t^2/2\sigma_t^2\right)\right]\left[\exp(2i\pi f_0 t)\right]$. Here, A is a normalizing factor equivalent to $1/\left(\sigma_t\sqrt{\pi}\right)^{1/2}$. The time-varying power of the signal around frequency f is the square modulus of the convolution: $P(t,f) = |w(t,f) * \mathrm{EEGsignal}(t)|^2$. The wavelet family used was defined by a ratio of 6 (f/σ_f), with a f range from 8 to 30 Hz and a 0.5 Hz resolution. The analysis was performed in the time interval from 1 to 7 s in steps of 10 ms. Power values for a 1 s long-time window, after 0.5 s of the MI cue, were selected and averaged over all the frequency band (8–30 Hz), as shown in Fig. 2. This time interval was chosen because previous studies showed that it is more likely to find MI patterns in the trial's time structure [32, 33]. Therefore, MI features for each EEG channel consisted of 120 RMI and 120 LMI vectors with 100 elements each (representative of time information) and with the averaged frequency features of the mu and beta band (representative of frequency information). The extracted features for each trial are contained in a matrix of 100 columns (time–frequency information) by 11 rows (EEG channels). All files were read and processed using the MATLAB® R2014b software from Mathworks and the free license toolbox FieldTrip [34].

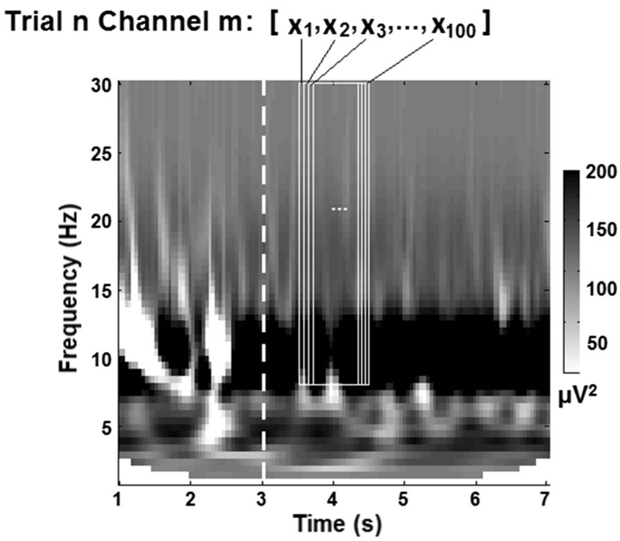

Fig. 2 Feature extraction procedure. Time–frequency representation for an EEG channel. *Dotted line* at 3 s indicates the motor imagery onset. *White square* shows the total area of extracted time–frequency information. *Smaller white rectangles* show the 10 ms time windows for which power is averaged from 8 to 30 Hz. Averaged power values for each window encompass the vector containing all the trial's features for this EEG channel $(x_1, x_2, x_3, \ldots, x_{100})$

2.5 Training and test subsets selection and cross-validation

Random subsampling was used for the segmentation of the extracted features in training and testing data. The training subset was randomly comprised of 60 RMI trials and 60 LMI trials, while the remaining data 60 RMI and 60 LMI trials comprised the testing subset. This process was repeated 30 times to satisfy the central limit theorem and to achieve a random subsampling cross-validation.

2.6 MI data normalization

Elements of each vector were normalized between 0 and 1. In order to achieve this, the maximum value of each feature vector of the training data was computed per channel. Afterward, both the training and the testing data feature vector elements were divided between the computed maximum values.

2.7 Spiking neuron model for MI classification with a time-varying input of EEG data (TVSN)

The IZ model was selected for the implementation of the neuron model since it offers a good tradeoff between biological accuracy and computational cost. The IZ model is described by two differential equations as [35]

$$v' = \frac{k(v - v_r)(v - v_t) - u + \boldsymbol{I}}{C} \tag{1}$$

$$\begin{aligned} &u' = a\{b(v - v_r) - u\} \\ &\text{if } v \geq v_{\text{peak}}, \text{then } v \leftarrow c, u \leftarrow u + d \end{aligned} \tag{2}$$

where v is the membrane potential, u is the recovery current, $\boldsymbol{I}$ is a vector with the input current arriving to the neuron, C is the membrane capacitance, v_r is resting membrane potential, v_t is the instantaneous threshold potential, k is the rheobase resistance, v_{peak} is the spike cutoff value, a is a recovery time constant, b is the input resistance, c is the voltage reset value and d is the outwards minus inwards currents during spike which affect the after-spike behavior of the model [35]. For the present work, a behavior of regular spiking neurons, which belong to the class 1 excitable neurons, is used for the SN model [35]. In order to achieve this, IZ parameters are set according to Table 1.

Euler's method was used to solve the two differential equations of the IZ model, since it provides a close approximation of the solution and offers low computational cost [35]. Euler's method was used with 100 time steps, so that a simulation time of 100 was used. The input of the neuron model was a time-varying current which encompasses the element-wise sum of the 100 power values computed for each channel. Each value was multiplied by a weight value (synaptic weights of the neuron model), 11 weight values were used, each one multiplying data from a separate EEG channel. In order to avoid low simulation input currents, a constant value of 100 is multiplied to the simulation input current, and this approach has been used before [23, 27]. The input current computation for the TVSN neuron model used for classification is described as

$$\boldsymbol{I}_{\text{TVSN}} = \sum_{i=1}^{11} k \cdot (w_i \cdot \boldsymbol{X}_i) \tag{3}$$

where $\boldsymbol{I}_{\text{TVSN}}$ is a vector of 100 elements containing the input simulation current of TVSN for each of the 100 time steps of the model's time window, w is a scalar with the synaptic weight of the i EEG channel, $\boldsymbol{X}$ is a vector of 100 elements comprised of EEG power values extracted from the i channel, k is a scalar with a value of 100. An illustration of the TVSN method is shown in Fig. 3.

Table 1 IZ parameters [35]

Parameter	Value
k	0.7
v_r	−60
v_t	−40
C	100
a	0.03
b	−2
c	−50
d	100
v_{peak}	35

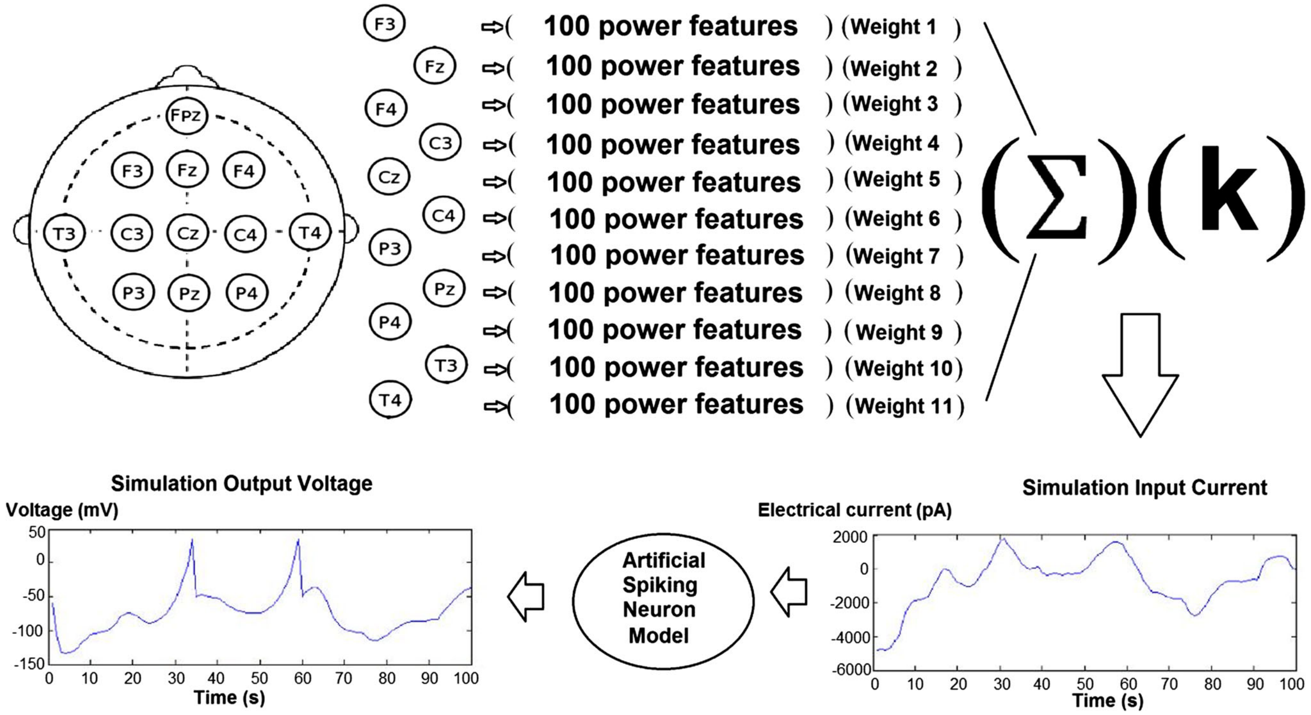
FPz
F3
Fz
F4
T3
C3
Cz
C4
T4
P3
Pz
P4
F3 ⇨(100 power features) (Weight 1)
Fz ⇨(100 power features) (Weight 2)
F4 ⇨(100 power features) (Weight 3)
C3 ⇨(100 power features) (Weight 4)
Cz ⇨(100 power features) (Weight 5)
C4 ⇨(100 power features) (Weight 6)
P3 ⇨(100 power features) (Weight 7)
Pz ⇨(100 power features) (Weight 8)
P4 ⇨(100 power features) (Weight 9)
T3 ⇨(100 power features) (Weight 10)
T4 ⇨(100 power features) (Weight 11)
(Σ)(k)
Simulation Input Current
Electrical current (pA)
2000
0
-2000
-4000
-6000
0 10 20 30 40 50 60 70 80 90 100
Time (s)
Artificial Spiking Neuron Model
Simulation Output Voltage
Voltage (mV)
50
0
-50
-100
-150
0 10 20 30 40 50 60 70 80 90 100
Time (s)

Fig. 3 Illustration of TVSN method. First EEG data are acquired, and then a feature vector containing 100 elements normalized between 0 and 1 are extracted for each channel. Each of the normalized vectors is multiplied by a weight. The resulting vectors are summed and afterward are multiplied by a scalar $k = 100$. The result is a single vector with 100 elements that is introduced as the input simulation current of the SN model. Spikes are counted from the output voltage of the simulation time window of the SN model and are related to a motor imagery label

MI is coded as time-related magnitude changes of spectral power; therefore, differences in the morphology of the changes across time can help to discriminate between RMI and LMI. These morphology differences across the two kinds of MI can elicit different numbers of spikes in the simulation output voltage of the model; consequently, a rate coding classification scheme was chosen. In the training phase of the SN, synaptic weights are adjusted, so that the Euclidian distance from the mean number of spikes elicited for RMI and LMI to the training vectors matches their class label.

Testing is done by transferring the synaptic weights values and averaged number of spikes elicited for both MI classes, computed in the training phase, to a SN model that classified the testing data. This SN model recognizes between RMI and LMI by computing the Euclidian distance of the testing data to each of the two averaged number of spikes related to the two MI classes. The class, to which the testing data is closest to, is the class of MI data that the model associates to the input data.

Synaptic weights are adjusted with the particle swarm optimization (PSO) algorithm. A detailed description of the PSO method can be found in [36], and it is described by the next equations:

$$\boldsymbol{v}_{i+1} = w\boldsymbol{v}_i + c_1\boldsymbol{r}_1(\text{PBest} - \boldsymbol{x}_i) + c_2\boldsymbol{r}_2(\text{GBest} - \boldsymbol{x}_i) \tag{4}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{v}_i \tag{5}$$

where i is the current generation of the swarm, $\boldsymbol{v}_i$ is a vector with the velocity of the particles, w is the inertia weight parameter, c_1 and c_2 are craziness factors for the local and global searches, respectively, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors whose values range from 0 to 1, PBest is the best position of each particle, GBest is the best position achieved so far by the whole particle swarm. $\boldsymbol{x}_i$ is a vector containing the position for each particle in the swarm. When using PSO for finding the best synaptic weights for solving the MI classification problem, the particle's position is a value for the weights. Since one weight is needed for each of the EEG channel's power, a total of 11 weights are needed, and thus $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ are vectors with 11 elements each. It has been demonstrated that a decreasing value of w is recommendable to boost PSO performance, so for our problem, the initial value of w is set to 1 and the final one to 0.1. c_1 and c_2 are both set to 1 in order to give the same priority to generalization and specialization of the solution. Boundaries for the search space of PSO also need to be set; for the present work, 100 and -100 are selected as upper and lower boundaries, respectively, while maximum and minimum particle velocities were set to 100 and -100, respectively. This means that the maximum value that each of the 11 weights can reach is 100 and the lowest is -100,

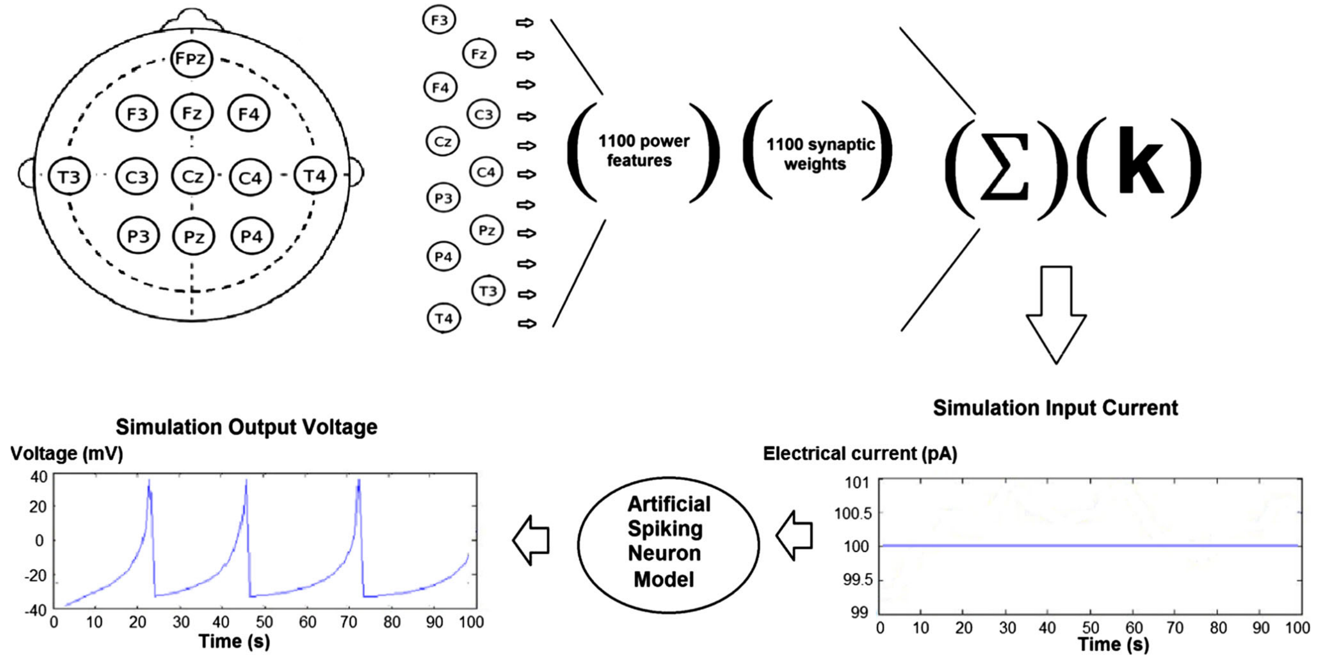

Fig. 4 Illustration of TUSN strategy. First, EEG data are acquired and then a feature vector containing 100 power values normalized between 0 and 1 are extracted from each channel. The 11 vectors of 100 elements each are concatenated as a single vector of 1100 elements. The resulting 1100 element vector is element-wise multiplied by a vector of 1100 weights. The elements of the resulting vectors are summed and the single value left after this addition is multiplied by a scalar $k = 100$. The result is a single value that is introduced as the input simulation current of the SN model. Number of spikes, which are related to a motor imagery label, is computed from the output voltage of the simulation time window of the SN model

before multiplication by k; after multiplication by k (Eq. 3), the maximum and minimum simulation current magnitudes are of 10,000 and $-10{,}000$, respectively. To train the SN models, 1000 iterations (generations) and 500 population members (particles) were used.

2.8 Spiking neuron model for MI classification with a time-unvarying input of EEG data (TUSN)

For the SN model in which the input is a time-unvarying simulation current, the SN parameters of the IZ model are the same as those for the TVSN. However, for this implementation, the simulation input current of the model is constant within the time window. Instead of 11 synaptic weights, 1100 synaptic weights are used, one weight for each of the extracted power features from each EEG channel. The computation of the input simulation current of the TUSN neuron model is described as

$$\boldsymbol{I}_{\mathrm{TUSN}} = \sum_{i=1}^{1100} (k * w_i * x_i) \tag{6}$$

where $\boldsymbol{I}_{\mathrm{TUSN}}$ is a scalar representative of the input simulation current for the TUSN. w is a scalar with the synaptic weight that multiplies the i power value. x is the i power value. k is a scalar equal to 100, which is multiplied by the input current in order to avoid low input simulation currents. As for the TVSN method, rate coding is also used for classification. The only difference for the training phase weight adjustment of the model, with respect to the one used in TVSN, is that the vectors $\boldsymbol{x}_i$, $\boldsymbol{v}_i$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ used by PSO are comprised of 1100 elements each. This approach produces a constant input simulation current for the SN model, as depicted in Fig. 4.

2.9 Comparison of performance of SN models and other well-established classification methods

In order to assess the performance of the SN models for MI classification, other well-established methods were also applied. The first of them was LDA, which is a method that transforms the original projection space of the classification problem, to a projection space that maximizes interclass variability and minimizes inner-class variability, in such a way that classes can be differentiated with a linear discriminant function. For the present work, the diagonal covariance matrix is used for LDA training, due to the high dimensionality of the problem. Another used classification method was a second generation artificial neural network (ANN), which discriminates between classes by multiplying the input values with synaptic weights and summing a variable known as bias. Afterward, the computed values are introduced to an inverse sigmoidal function, the activation function of the chosen second generation neuron models, in order to map the outputs in values ranging from 0 to 1. The ANN was trained with the backpropagation algorithm, with its stopping criteria set to 1000 iterations. The network architecture was configured as a three-layer network, with 1100 neurons in the input layer, 10 neurons in the hidden layer and 1 neuron in the output layer. Each input neuron was fed by 1 of the 1100 power features. The function of the input layer was to accept the input into the network, with no operations performed to the input. Each neuron in the hidden layer was fed with each of the 1100 neuron outputs of the input layer. Each of the hidden layer neuron outputs were connected to the output layer neuron. In addition, classification was also performed with a support vector machine (SVM), which is a robust classification method that uses some of the training samples (the support vectors) to build a discriminant space from which the classes can be more easily differentiated. A SVM can work with different kernels to establish the discriminant function; in the present work, two types of SVM kernels were used, the linear one (SVMLIN) and the radial basis function one (SVMRBF). SVMLIN and SVMRBF were trained with the sequential minimal optimization algorithm proposed by Platt in 1998, with no auto scaling, with 0% of the variables allowed to violate Karush–Kuhn–Tucker conditions of SMO, with box constraint of the soft margin set to 1, and with stopping criteria of 1500 iterations. SVMRBF sigma parameter was set to 1. All of the classification methods were trained and tested with the same subsets as the SN models.

With the purpose of comparing the performance of the implemented classification methods, percentage of correct classification (%CC) was computed for each method. This was done by averaging the %CC achieved by each of the 30 experiments per subject, obtained using a random subsampling cross-validation stage as explained on Sect. 2.5. The practical level of chance for each experiment, which indicates if the %CC is significantly higher than would be expected by chance, was computed by using the criteria stated in [37].

A two-way analysis of variance (ANOVA) was performed for assessing significant differences ($\alpha = 0.05$) between the TVSN method and each of the other classification methods, tested with data from the patient's group. Another two-way ANOVA was performed for the healthy subject's group. Additionally, multiple comparisons were computed using a Bonferroni correction ($\alpha = 0.05$) to determine differences in %CC between classification methods for each subject or patient.

The computational cost of each method, which is described in terms of the arithmetical operations needed for the implementation of their testing stages, and their computational complexity, described by using Big-O notation, were estimated. The training stage computational cost and complexity were not taken into account since most BCIs designed for stroke rehabilitation perform the training or adjustment processes of the system offline.

3 Results

3.1 Performance of classification methods

Results of patient's MI classification for all evaluated methods are shown in Table 2. TVSN classified MI from 5 patients above the practical level of chance. The only other method that showed a similar performance for such number of patients was the ANN. For the patient's group TVSN had significant ($p < 0.05$) higher %CC than TUSN, SVMRBF and LDA methods. No significant differences were observed between TVSN with SVMLIN and ANN. For none of the methods, TVSN had a significant ($p < 0.05$) lower performance.

Table 3 shows the results of the ANOVA for multiple comparisons with Bonferroni correction, between the %CC

Table 2 Classification results for stroke patients

	TVSN	TUSN	SVMLIN	SVMRBF	LDA	ANN
Patient 1	**79.6 (±5)**	**73.9 (±4)**	**79.9 (±2)**	49.7 (±1)	**71.2 (±7)**	**78.2 (±6)**
Patient 2	**62.7 (±6)**	53.7 (±3)	**58.5 (±4)**	51.6 (±1)	**57.4 (±3)**	**57.1 (±4)**
Patient 3	**58.1 (±7)**	**57.6 (±4)**	**56.3 (±6)**	**58.1 (±6)**	49 (±6)	**57.1 (±6)**
Patient 4	49.9 (±5)	53.1 (±3)	54.8 (±4)	51.1 (±4)	51.9 (±4)	53.8 (±4)
Patient 5	**56 (±4)**	**56.2 (±3)**	55.4 (±3)	54.4 (±3)	**60.8 (±3)**	**58.6 (±4)**
Patient 6	**61.8 (±4)**	**57.5 (±6)**	**64.3 (±2)**	**60.9 (±2)**	**59 (±3)**	**58.6 (±5)**
Mean	61.4 (±10)	58.7 (±8)$^{+}$	61.5 (±10)$^{=}$	54.3 (±4)$^{+}$	58.2 (±8)$^{+}$	60.5 (±9)$^{=}$

Mean (±Standard Deviation) expressed in percentages

%CC above the practical level of chance (56%) are marked in bold. "+" indicates that TVSN was significantly ($p < 0.05$) higher than the compared method, "−" that it was significantly ($p < 0.05$) lower, and "=" that the compared method's performance showed no significant difference

Table 3 Comparison between TVSN method's performances against other tested classification methods for each patient's data

	TUSN		SVMLIN		SVMRBF		LDA		ANN	
Patient 1	<0.001	+	1.000	=	<0.001	+	<0.001	+	1.000	=
Patient 2	<0.001	+	0.006	+	<0.001	+	<0.001	+	<0.001	+
Patient 3	1.000	=	1.000	=	1.000	=	<0.001	+	1.000	=
Patient 4	0.091	=	<0.001	−	1.000	=	1.000	=	0.013	−
Patient 5	1.000	=	1.000	=	1.000	=	<0.001	−	0.527	=
Patient 6	0.004	+	0.460	=	1.000	=	0.261	=	0.109	=

"+" symbol indicates that TVSN was significantly higher than the compared method, "−" that it was significantly lower, and "=" that the compared method's performance shows no significant difference. The computed *p* value for each comparison is also shown

of TVSN and the ones computed for other classification methods with data from each patient. For 3 patients, TVSN showed a significant higher performance ($p < 0.05$) compared to TUSN. For data from patient 2, TVSN showed a significant higher performance ($p < 0.05$) than all other tested methods. For patient 4, SVMLIN and ANN showed significant higher performances ($p < 0.05$) than TVSN. For patient 5, LDA showed a significant higher performance ($p < 0.05$) than TVSN. For data from the remaining patients, TVSN classification performance was significant higher ($p < 0.05$) or no significant differences were found with the other tested classification methods.

Results of healthy subject's MI classification for all evaluated methods are shown in Table 4. TVSN, ANN and LDA had a performance above the practical level of chance for 3 subjects. SVMLIN was the classification method that achieved performances above the practical level of chance for the highest number of subject's MI data (4 subjects). For TUSN and SVMRBF, only one subject was classified above the practical level of chance. For the healthy subject's group, TVSN had significant higher performance ($p < 0.05$) than TUSN and SVMRBF. No significant differences were observed between TVSN with SVMLIN, ANN and LDA. For none of the methods, TVSN had a significant lower performance ($p < 0.05$).

Table 5 shows the results of the ANOVA for multiple comparisons with Bonferroni correction, between the %CC of TVSN and the ones computed for other classification methods with data from each healthy subject. For 1 healthy subject, TVSN showed a significant higher performance ($p < 0.05$) than TUSN. For subject 6, SVMLIN showed a significant higher performance ($p < 0.05$) than TVSN. For subjects 1 to 5, TVSN classification performance was significant higher ($p < 0.05$), or no significant differences were found with the other tested classification methods.

At least one of the tested classification methods achieved a %CC above the practical level of chance for 5 out of 6 stroke patients and for 5 out 6 healthy subject's data. These results coincide with the assumption that BCI illiteracy (classification percentages below the practical level of chance) cannot be completely eliminated [38].

3.2 Computational cost and complexity of the assessed classification methods

For the LDA method to recognize between RMI and LMI patterns a linear discriminant function needs to be solved as

$$f(class) = \begin{cases} \mathrm{RMI}, \mathrm{XL} + k < 0 \\ \mathrm{LMI}, \mathrm{XL} + k \geq 0 \end{cases} \tag{7}$$

Table 4 Classification results for healthy subjects

	TVSN	TUSN	SVMLIN	SVMRBF	LDA	ANN
Subject 1	**75 (±6)**	**73.2 (±2)**	**77.5 (±4)**	**59.1 (±7)**	**73.8 (±5)**	**73.9 (±8)**
Subject 2	**63.3 (±6)**	55.4 (±3)	**60.3 (±5)**	50.7 (±3)	**61.2 (±7)**	55.5 (±4)
Subject 3	**58.7 (±4)**	55.8 (±4)	**58.7 (±3)**	50 (±1)	52.4 (±5)	**56.2 (±3)**
Subject 4	55.7 (±5)	54.3 (±5)	**57.8 (±6)**	54 (±4)	55.3 (±3)	53.7 (±3)
Subject 5	55.7 (±4)	54.8 (±6)	54.5 (±3)	50 (±1)	**58.5 (±4)**	**58.4 (±5)**
Subject 6	49.3 (±4)	50.6 (±4)	53.6 (±5)	50 (±1)	49.1 (±5)	52.1 (±4)
Mean	59.6 (±8)	57.3 (±8)$^{+}$	60.4 (±9)$^{=}$	52.3 (±4)$^{+}$	58.3 (±9)$^{=}$	58.3 (±8)$^{=}$

Mean (±Standard Deviation) expressed in percentages

%CC above the practical level of chance (56%) are marked in bold. "+" indicates that TVSN was significantly ($p < 0.05$) higher than the compared method, "−" that it was significantly ($p < 0.05$) lower, and "=" that the compared method's performance showed no significant difference

Table 5 Comparison between TVSN method's performances against other tested classification methods for healthy subject's data

	TUSN		SVMLIN		SVMRBF		LDA		ANN	
Subject 1	1.000	=	0.512	=	<0.001	+	1.000	=	1.000	=
Subject 2	<0.001	+	0.161	=	<0.001	+	1.000	=	<0.001	+
Subject 3	0.190	=	1.000	=	<0.001	+	<0.001	+	0.411	=
Subject 4	1.000	=	0.570	=	1.000	=	1.000	=	1.000	=
Subject 5	1.000	=	1.000	=	<0.001	+	0.225	=	0.273	=
Subject 6	1.000	=	0.004	−	1.000	=	1.000	=	0.290	=

"+" symbol indicates that TVSN was significantly higher than the compared method, "−" that it was significantly lower, and "=" that the compared method's performance shows no significant difference. The computed p value for each comparison is also shown

where $\boldsymbol{X}$ is a vector of 1100 power values across the 11 EEG channels. $\boldsymbol{L}$ is a vector of 1100 elements with the linear coefficients of the boundary equation. k is the constant term of the boundary equation. k is a scalar. Both $\boldsymbol{X}$ and $\boldsymbol{L}$ are computed in the training phase of the classification method. A total of 1100 multiplications and 1101 additions are needed to be performed for the LDA to recognize between MI patterns.

Since the kernel trick allows for different discrimination functions to be used for a classification method without a significant increase in the computational cost, both SVMRBF and SVMLIN computational costs can be described as

$$f(class) = \begin{cases} \text{RMI}, \sum_{1}^{j}(\alpha_j k(S_j, X) + b) < 0 \\ \text{LMI}, \sum_{1}^{j}(\alpha_j k(S_j, X) + b) \geq 0 \end{cases} \quad (8)$$

where $\boldsymbol{S}$ is the j support vector which has 1100 elements, k is the kernel used, α is the j weight of the support vector, $\boldsymbol{X}$ is the input vector which has 1100 elements and b is the constant of the discriminant function. α and b are scalars. For SVMLIN, the kernel function is equivalent to the product of $\boldsymbol{S}_j$ and $\boldsymbol{X}$. This property allows to multiply α and $\boldsymbol{S}$ in order to obtain a vector which can be directly multiplied by the input $\boldsymbol{X}$, and therefore, computational cost for the SVMLIN is 1100 multiplications and 1101 additions. The SVMRBF method can have different computational costs depending on the number of support vectors set in the algorithm's training phase. The radial basis function kernel used for SVMRBF is described as

$$k(\boldsymbol{S}_j, \boldsymbol{X}) = e^{-\left(\left\|S_j - X^2\right\|/2\sigma^2\right)} \quad (9)$$

where $\boldsymbol{S}_j$ is the j support vector which has 1100 elements, $\boldsymbol{X}$ is the input vector which has 1100 elements, σ is the sigma parameter of the radial basis function which is set to 1. σ is a scalar. Equation 9 is comprised by 2200 additions or subtractions and by 1104 multiplications or divisions (for the sake of simplicity the exponential function is estimated to be comprised by 1100 multiplications). Therefore, SVMRBF computational cost will be of 2200 additions and 1104 multiplications, plus 1 addition and 1 multiplication (from Eq. 8) times the number of support vectors. In order to compute an approximation of these costs, the average number of support vectors (s) set by the training phases of the SVMRBF was computed for all of the 30 experiments of each subject, and these results are shown in Table 6.

The ANN model used in this work had 1100 input neurons, 10 hidden neurons and 1 output neuron. The input layer directly feeds the 1100 extracted features into each one of the 10 hidden neurons, and thus, input neurons have no computational cost. Each hidden neuron output was connected to a single neuron located in the output layer. The ANN solves the following discrimination functions in order to classify between RMI and LMI as

$$l_i = \frac{1}{1 + e^{-\left(\sum_{i=1}^{1100}(x_i v_i + b)\right)}} \quad (10)$$

where l_i is the output of the i neuron from the network's hidden layer. x is the i feature, v is the synaptic weight related to the i feature, b is the bias of the hidden layer neurons. l_i, x, v and b are scalars.

Table 6 Computational costs of SVMRBF

Subject	SVMRBF		
	α	Additions	Multiplications
Patient 1	120	264,120	132,600
Patient 2	119	261,919	131,495
Patient 3	104	228,904	114,920
Patient 4	119	261,919	131,495
Patient 5	116	255,316	128,180
Patient 6	114	250,914	125,970
Subject 1	103	226,703	113,815
Subject 2	120	264,120	132,600
Subject 3	120	264,120	132,600
Subject 4	111	244,311	122,655
Subject 5	120	264,120	132,600
Subject 6	120	264,120	132,600

$$f(class) = \begin{cases} \text{RMI}, \dfrac{1}{1 + e^{-\left(\sum_{i=1}^{10}(l_i w_i + d)\right)}} < 0 \\ \text{LMI}, \dfrac{1}{1 + e^{-\left(\sum_{i=1}^{10}(l_i w_i + d)\right)}} \geq 0 \end{cases} \tag{11}$$

where l_i is the output of the i neuron from the network's hidden layer. w is the synaptic weight that multiplies the output data from the i neuron, d is the bias of the output layer neuron. l_i, w and d are scalars. Therefore, the ANN model requires a total of 11,010 multiplications and 11,010 additions to classify the MI patterns if the exponential function, the single addition and single division required for computing the activation function are left out from the calculation. This is justified since they are only performed once for each artificial neuron.

Both TVSN and TUSN models implementations first require computing the input simulation current of the model; this is achieved by solving Eqs. 3 and 6, respectively. This accounts for 1101 multiplications and 1100 additions for both approaches. The computational cost of solving the IZ model with Eulerś method totals 7 multiplications and 8 additions per step of the simulation time window. For the 100 step simulation time window used in this application, a total of 700 multiplications (taking into account divisions as multiplications) and 800 additions (taking into account subtractions as additions) are needed. This makes the computational cost for classifying MI data with both TVSN and TUSN neuron models of 1801 multiplications and 1900 additions.

Table 7 shows both the computational cost and complexity of implementing each of the classification methods. The classification methods with the lower implementation computational costs were LDA and SVMLIN. Compared to LDA and SVMLIN, TVSN and TUSN have the second lowest computational costs with 1.63 times the cost of LDA and SVMLIN, followed by the ANN with a factor of almost 10 times, and SVMRBF with a factor of 170. Regarding computational complexity of the methods, the last column of Table 7 shows that LDA and SVMLIN computational complexity increases linearly with the input data size. TVSN and TUSN computational complexity states that computational cost linearly increases with the input data size plus the computational cost of solving Izhikevich model with Eulerś method, which is constant, and can be stated to be approximately equal to LDA's and SVMLIN's. The computational complexity of SVMRBF is dependent on the input's data size and the number of support vectors used for computing the classifier's output. ANN computational complexity is dependent on the input size and the number of neurons in the different layers of the network.

4 Discussion

The fact that TVSN showed better classification performances than TUSN is likely to be the result of the preservation of time–frequency information. This information is important for classification since it is well known that EEG is a non-stationary signal. Additionally, the search space of TVSN (11 weights to be adjusted) is significantly lower than the required for TUSN (1100 weights to be adjusted), which increased the possibility for TUSN training to reach inefficient local solutions. The TVSN method tested in the present work does not loose generalization properties with training since it does not affect the time–frequency structures presented by each EEG channel. This feature could make this type of SN models less susceptible to overfitting. The regular spiking behavior of the spiking neuron models tested in this work resembles a linear transfer function, since spike rate is proportional to the simulation's input current magnitude. Even though both TVSN and TUSN share the same spiking neuron behavior, TVSN's transfer function is more complex than a linear transfer function since it incorporates the time structure of its simulation input current. Therefore, a SN classification

Table 7 Computational cost and complexity of implementation stage of classification methods

Methods	Computational cost		Computational complexity	
	Number of additions	Number of mutiplications	Normalized computational cost (with respect to LDA)	Big-O notation
TVSN	1900	1801	O(1.63)	$O(n + c) \approx O(n)$
TUSN	1900	1801	O(1.63)	$O(n + c) \approx O(n)$
SVMLIN	1101	1100	O(1)	$O(n)$
SVMRBF	254,215	127,627	O(170)	$O(nk)$
LDA	1101	1100	O(1)	$O(n)$
ANN	11,010	11,010	O(10)	$O(nk + d)$

N number of features, *c* computational cost of solving Izhikevich (IZ) model with Eulerś method, *k* number of neurons or support vectors, and *d* cost of computing the ANN output layer method that preserves time–frequency information is likely to have a better hand MI classification performance, or in the worst case a similar one, than one that does not preserve it. Independently of the number of EEG channels used by a BCI, TVSN is a better choice for hand MI classification than TUSN for two reasons: (1) The computational requirements are the same in both cases and (2) TVSN classification performance is better. Previous studies which evaluate SN models for EEG classification do not take into account the advantages of implementing a time-varying simulation current, like TVSN. This approach could significantly improve classification performances of such models without increasing their computational cost. Additionally, classification of both TVSN and TUSN models could be improved by tuning the parameters of Izhikevich model, since it has been reported that a single spiking neuron classification performance can be increased, after the model's parameters are fine-tuned [23].

This study shows that TVSN achieved similar classification performances than the tested ANN, and this suggest that SN models have a higher computational power than the AN models. This premise is reinforced by the fact that TVSN is comprised of a single neuron model, while three layers are used for the ANN. TVSN computational cost advantages are seen for both stroke patients and healthy subjects MI classification. This is important because it suggests that a third generation model should be used instead of a second generation one for MI classification in BCI applications, only if the SN uses a time-varying current in its model.

For BCI developers, one of the most interesting results shown in this study is the comparison done between MI classification performances of LDA and TVSN. LDA is robust and yet computational inexpensive to implement in a BCI; therefore, it is one of the most used classification methods in BCIs [33, 39–41]. Hence, the fact that TVSN showed higher classification performances than LDA for stroke patients is an important observation. The most probable cause for this is that LDA does not preserve time–frequency features. On the other hand, most of the healthy subjects MI classification using LDA showed similar performances as with TVSN. Therefore, LDA could be implemented for BCI designs aimed for healthy populations due to its lower implementation computational cost and good classification performance, whereas SN models could be used for BCIs aimed for stroke rehabilitation using MI. One advantage of SN models over LDA is that the mathematical restrictions of computing covariance matrixes are not applied to them; therefore, convergence is guaranteed with an SN model.

One of the main advantages of SVMs over ANNs is that they are far less susceptible to overfitting. On the other hand, ANNs have a fixed architecture, while some SVMs implementation architectures are size variable (set up during training) and in most cases are computationally more expensive. In this work, the SVMLIN showed a much better performance than SVMRBF. The main reason for this could be that the dimensionality of the search space, added to a two class problem, contributed to a better performance of the linear discrimination function. This has also been reported for other classification problems [42]. TVSN clearly showed a better performance than SVMRBF for hand MI classification of both healthy subjects and stroke patients. However, for most healthy subjects and stroke patients, SVMLIN showed performances similar to the ones achieved with TVSN. There is one exception, a stroke patient for which TVSN was better. This shows that TVSN can be used as the classification stage of a BCI system based on MI. Since it will have a performance at least as good as that of a SVMLIN, there is a possibility that TVSN can have better performance than SVMLIN for some patient's data. Compared to SVMLIN, the computational cost of TVSN was higher; however, its computational complexity was almost the same. Therefore, the implementation computational cost of TVSN will be closer to that of SVMLIN if more features are used as inputs for classification, for example a higher number of EEG channels. TVSN can be implemented with fewer system requirements, fielding a much lower computational cost than a SVMRBF, and also having a lower computational complexity.

TVSN model only involves one spiking neuron; therefore, it is probable that a neural network comprised of these models can increase classification accuracies. This has been observed for EEG data classification using SNN architectures, like the NeuCube, which performance was higher than the ones obtained with SVM for mental tasks classification [28]. The NeuCube is able to perform classification of EEG data by first feeding it into an input stage, which can use different methods comprised of spiking neurons to transform them into a sequence of spikes [43]. On the other hand, TVSN is fed with power features extracted with WT. Therefore, the NeuCube has the advantage that no prior feature extraction stage is required for classification. This work shows that feeding power features into an SN model is also a good strategy for EEG data classification. Furthermore, TVSN could be implemented as part of the input module of the NeuCube, or other architectures comprised of SNN. Time code is used by the SN models tested in the input module of the NeuCube, this means that TVSN, which uses rate code, could be used as an alternative to these input SN models [43]. Another stage of the NeuCube is comprised by an evolving SNN, which adds or eliminates neurons from a 3D mapped model; therefore, spatial information could also assist the NeuCube in achieving good classification performances.

However, TVSN has shown good classification performances with only one neuron and could increase the computational power of this stage of the NeuCube, if added to the models´ 3D architecture. The NeuCube has multiple parameters that need to be optimized, whereas TVSN only requires the optimization of the model's synaptic weights. This could imply that TVSN can be set up more easily to achieve competitive classification performances. But if the NeuCube's setup is optimal, classification performance would be likely higher than TVSN's. TVSN computational cost and complexity are lower than those of the NeuCube and make it suitable for online BCI implementations. However, in the NeuCube, information is mapped in such a way that it allows to perform a deep analysis of the input EEG data and provide neurofeedback, giving both models complementary features.

This study shows that the tested classification methods can have significant higher or lower performances for each BCI user. Therefore, TVSN is a further addition to the possible classification methods that can have a good performance and low computational cost, for healthy subject's and patient's MI classification. Data of each BCI user should be tested in order to select the most suitable classification method. Additionally, adding TVSN as part of a classifier assemble could allow more patients and healthy subjects to be able to control a BCI.

The main disadvantage of TVSN and TUSN compared to all other tested methods is that the computational cost required for its training phase is extensive. In order to further optimize the training phase of SN models applied to MI classification, other training schemes must be tested, like ABC, STDP and gradient-descent-based methods. These training scheme comparisons should be made with a SN model that incorporates a time-varying simulation current, like the TVSN model tested in the present work. In this work spectral features were used for classification, although other feature extraction algorithms could also be tested. A limitation of this study is the small sample of patients and healthy subjects; however, this is partially compensated by a large sample of MI information acquired for each participant. Also most works that evaluate MI classification performance of SN models do not use information acquired from stroke patients, which increases the relevance of the present study.

5 Conclusion and future work

The present study shows that SN models can achieve competitive performances of hand MI classification extracted from EEG. This was achieved with a fraction of the implementation computational cost of other methods such as ANNs and SVMRBFs. However, in order to achieve such performances, preservation of the time–frequency features was required. Preservation can be performed with a time-varying simulation current with respect to the simulation time of the SN model. SN classification performances are obtained with data from both stroke patients and healthy subjects, which imply that SN models could increase the performance of BCI systems. This could allow more people to use these kinds of systems for applications ranging from neurorehabilitation to simple communication with daily electronic devices. The low computational cost of the SN models tested in this work allows them to be implemented as a BCI classification stage without increasing the hardware requirements of the system, which means that a more portable and accessible BCI system can be designed. However, further studies are necessary in order to confirm these findings. These studies should optimize the parameters of the spiking neuron models, in order to test if other neuron behaviors can increase classification performance. More stroke patients should be recruited, in order to further test the SN models with a larger sample of MI information. Online tests should also be performed, and different training approaches for the SN models could also be explored in order to reduce the training computational cost of the model. A single SN is used for this work; however, a network of SN could increase even more MI classification performances of BCI systems, and the SN approach of this work could be used to increase the performance of SNN architectures. The next step of our work is the design of a SNN comprised of TVSN models. There is still a long way ahead to achieve a clinical BCI system capable of providing neurorehabilitation for stroke patients; however, it seems that SN models have the potential to bring these kinds of systems closer to reality.

Acknowledgements The authors would like to thank Consejo Nacional de Ciencia y Tecnología and Universidad La Salle for the economic support under Grant SALUD-2015-2-262061 and NEC-03/15, respectively.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mozaffarian D, Benjamin EJ, Go AS et al (2015) Hearth disease and stroke statistics 2015 update a report from the American Heath Association. Circulation 131:29–322
2. Rodriguez M, Llanos C, Sabate M (2009) The kinematicks of motor imagery: comparing the dynamics of real and virtual movements. Neuropsychologia 47(2):489–496
3. Carrillo-de-la-peña MT, Galdo-Alvarez S, Lastra-Barreira C (2008) Equivalent is not equal: primary motor cortex (MI)

activation during motor imagery and execution of sequential movements. Brain Res 1226:134–143

4. Kraeutner S, Gionfriddo A, Bardouille C, Boe S (2014) Motor imagery-based brain activity parallels that of motor execution: evidence from magnetic source imaging of cortical oscillations. Brain Res 1588:81–91
5. Pfurtscheller G, Lopes da Silva FH (1999) Event-related EEG/EMG synchronization and desynchronization: basic principles. Clin Neurophysiol 110(11):1842–1857
6. Nguyen T, Khosravi T, Creighton D, Nahavandi S (2015) Fuzzy system with tabu search learning for classification of motor imagery data. Biomed Signal Proces 20:61–70
7. Muller-Putz GR, Kaiser V, Solis-Escalante T, Prurtscheller G (2010) Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG. Med Biol Eng Comput 48:229–233
8. Nguyen T, Khosravi A, Creighton D, Nahavandi S (2015) EEG data classification using wavelet features selected by Wilcoxon statistics. Neural Comput Appl 26:1193–1202
9. Ahangi A, Karamnejad M, Mohammadi N, Ebrahimpour R, Bagheri N (2013) Multiple classifier system for EEG signal classification with application to brain–computer interfaces. Neural Comput Appl 23:1319–1327
10. Xanthopoulos P, Pardalos PM, Trafalis TB (2013) Linear discriminant analysis. In: Xanthopoulos P, Pardalos PM, Trafalis TB (eds) Robust data mining, 1st edn. Springer, Berlin, pp 27–33
11. Xanthopoulos P, Pardalos PM, Trafalis TB (2013) Support vector machines. In: Xanthopoulos P, Pardalos PM, Trafalis TB (eds) Robust data mining, 1st edn. Springer, Berlin, pp 35–48
12. Xingui H, Shaohua X (2010) Artificial neural networks. In: Xingui H, Shaohua X (eds) Process neural networks, theory and applications. Springer, Berlin, pp 20–42
13. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117:500–544
14. Gerstner W, Kistler WM (2002) Formal spiking neuron models. In: Gerstner W, Kistler WM (eds) Spiking neuron models. Cambridge University Press, Cambridge, pp 94–97
15. Izhikevich EM (2003) Simple model of spiking neurons. IEEE T Neural Networ 14:1569–1572
16. Xu Y, Zeng X, Lixin H, Jing Y (2013) A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. Neural Netw 43:99–113
17. Bohte SM, Kok JN, La Poutré H (2002) Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48:17–37
18. Yu Q, Tang H, Tan KC, Yu H (2014) A brain-inspired spiking neural network model with temporal encoding and learning. Neurocomputing 138:3–13
19. Gutig R, Sompolinsky H (2006) The tempotron a neuron that learns spike timing based decisions. Nat Neurosci 9:420–428
20. Polunak F, Kasinski A (2010) Supervised learning in spiking neural networks with resume: sequence learning, classification and spike shifting. Neural Comput 22(2):467–510
21. Wang J, Belatreche A, Maguire L, McGinnity TA (2014) An online supervised learning method for spiking neural networks with adaptive structure. Neurocomputing 144:526–536
22. Kasabov NK (2014) NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. Neural Netw 52:62–76
23. Cachon A, Vázquez RA (2015) Tuning the parameters of an integrate and fire neuron via a genetic algorithm for solving pattern recognition problems. Neurocomputing 148:187–197
24. Garro BA, Rodriguez K, Vazquez RA (2016) Classification of DNA microarrays using artificial neural networks and ABC algorithm. Appl Soft Comput 38:548–560
25. Vazquez RA, Garro BA (2015) Training spiking neural models using artificial bee colony. Comput Intel Neurosci. doi:10.1155/2015/947098
26. Garro BA, Vázquez RA (2015) Designing artificial neural networks using particle swarm optimization algorithms. Comput Intel Neurosci. doi:10.1155/2015/369298
27. Kampakis S (2012) Improved Izhikevich neurons for spiking neural networks. Soft Comput 16(6):943–953
28. Kasabov N, Capecci E (2015) Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes. Inform Sci 294:565–575
29. Chen Y, Hu J, Kasabov N, Hou Z, Cheng L (2013) NeuCubRehab: a pilot study for EEG classification in rehabilitation practice based on spiking neural networks. In: Lee M, Hirose A, Hou ZG, Kil RM (eds) Neural information processing. Springer, Berlin, pp 70–77
30. Pfurtscheller G, Nueper C (2001) Motor imagery and direct brain–computer communication. Proc IEEE 89(7):1123–1134
31. Hjort B (1975) An on-line transformation of EEG scalp potentials into orthogonal source derivations. Electroen Clin Neuro 39:526–530
32. Cantillo-Negrete J, Gutierrez-Martinez J, Flores-Rodriguez TB, Carino-Escobar RI, Elias-Vinas D (2014) Characterization of electrical brain activity related to hand motor imagery on healthy subjects. Rev Invest Clin 66(S1):111–121
33. Cantillo-Negrete J, Gutíerrez-Martínez J, Carino-Escobar RI, Carrillo-Mora P, Elias-Viñas D (2014) An approach to improve the performance of subject-independent BCIs-based on motor imagery allocating subjects by gender. BioMed Eng OnLine. doi:10.1186/1475-925X-13-158
34. Oostenveld R, Fries P, Eric M, Shoffelen J-M (2011) FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. Comput Intel Neurosci. doi:10.1155/2011/156869
35. Izhikevich EM (2007) Dynamical systems in neuroscience: the geometry of excitability bursting. MIT Press, Cambridge
36. Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of the IEEE international conference on evolutionary computation, pp 69–73
37. Muller-Putz JR, Sherer R, Brunner C, Leeb R, Pfurtscheller G (2008) Better than random? A closer look on BCI results. Int J Bioeletromagn 10:52–55
38. Edlinger G, Allison BZ, Guger C (2014) How many people can use a BCI system? In: Kansaku K, Cohen LG, Birbaumer N (eds) Clinical systems neuroscience. Springer, Berlin, pp 33–66
39. Choi D, Ryu Y, Lee Y, Lee M (2011) Performance evaluation of a motor-imagery-based EEG-Brain computer interface using a combined cue with heterogeneos training data in BCI-Naïve subjects. BioMed Eng Online 10(91):1–12
40. Rodríguez-Bermúdez G, García-Laecina P (2012) Automatic and adaptative classification of electroencephalographic signals for brain computer interfaces. J Med Syst 36(1):51–63
41. Li Y, Koike Y (2011) A real-time BCI with a small number of channels based on CSP. Neural Comput Appl 20:1187–1192
42. Song S, Zhan Z, Long Z, Zhang J, Yao L (2011) Comparative study of SVM methods combined with voxel selection for object category classification on fMIR data. PLoS ONE. doi:10.1371/journal.pone.0017191
43. Kasabov N, Scott N, Tu E, Marks S, Sengupta N, Capecci E, Othman M, Doborjeh M, Murli N, Hartono R, Espinosa-Ramos J, Zhou L, Alvi F, Wang G, Taylor D, Gulyaev S, Mahmoud M, Hou ZG, Yang J (2016) Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: design methodology and selected applications. Neural Networks 78:1–14