**TOPICAL REVIEW**

# Support vector machines to detect physiological patterns for EEG and EMG-based human–computer interaction: a review

IOP Publishing Journal of Neural Engineering
J. Neural Eng. **14** (2017) 011001 (27pp) doi:10.1088/1741-2552/14/1/011001

Topical Review

# Support vector machines to detect physiological patterns for EEG and EMG-based human–computer interaction: a review

**L R Quitadamo[1,2,4], F Cavrini[1], L Sbernini[1], F Riillo[1], L Bianchi[3], S Seri[2] and G Saggio[1]**

[1] Department of Electronic Engineering, University of Rome Tor Vergata, Rome, Italy
[2] School of Life and Health Sciences, Aston Brain Center, Aston University, Birmingham, UK
[3] Department of Civil Engineering and Computer Science, University of Rome Tor Vergata, Rome, Italy

E-mail: l.quitadamo@aston.ac.uk

Received 23 July 2015, revised 21 October 2016
Accepted for publication 7 November 2016
Published 9 January 2017


**Abstract**

Support vector machines (SVMs) are widely used classifiers for detecting physiological patterns in human–computer interaction (HCI). Their success is due to their versatility, robustness and large availability of free dedicated toolboxes. Frequently in the literature, insufficient details about the SVM implementation and/or parameters selection are reported, making it impossible to reproduce study analysis and results. In order to perform an optimized classification and report a proper description of the results, it is necessary to have a comprehensive critical overview of the applications of SVM.

The aim of this paper is to provide a review of the usage of SVM in the determination of brain and muscle patterns for HCI, by focusing on electroencephalography (EEG) and electromyography (EMG) techniques. In particular, an overview of the basic principles of SVM theory is outlined, together with a description of several relevant literature implementations. Furthermore, details concerning reviewed papers are listed in tables and statistics of SVM use in the literature are presented. Suitability of SVM for HCI is discussed and critical comparisons with other classifiers are reported.

Keywords: support vector machines, human–computer interaction, EEG, EMG, brain–computer interface

(Some figures may appear in colour only in the online journal)

## 1. Introduction

Human–computer interaction/interface (HCI) and human-machine interface (HMI) consist of technologies that allow humans to control external peripherals or electronic devices. In order to achieve such control, humans can either interact with the devices by means of a 'direct' medium, such as vision, hearing, touch, and gesture [1–4], or an 'indirect' one, such as brain or muscular activity. The indirect activity can be measured using techniques, for example ElectroEncephaloGram (EEG) and ElectroMyoGrams (EMG), which furnish data to be converted into commands for the peripherals.

[4] Author to whom any correspondence should be addressed. School of Life and Health Sciences, Aston University, B4 7ET, Birmingham, UK.

 

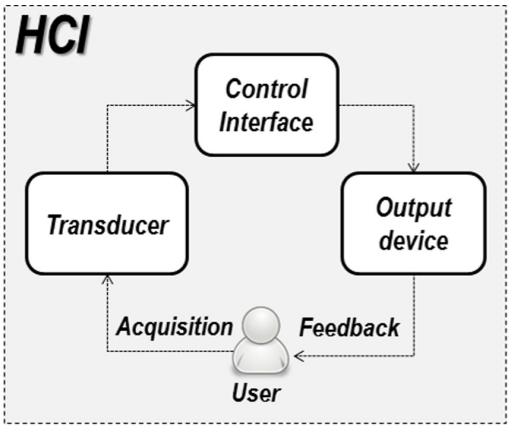


**Figure 1.** Functional model of a generic HCI system.

Specifically, EEG and EMG play a fundamental role in HCI, the main source of data for driving electronic/electromechanical devices being used to support disables' life routines and rehabilitation. In particular, EEG can be the core of a special interface, that is, the brain–computer interface (BCI) used by people who are paralyzed after traumas or degenerative diseases [5, 6]. In addition, EMG can be the source signal for amputees to drive prostheses, artificial limbs or exoskeletons, so to recover missed limb functionalities by using residual muscles activity [7, 8].

Physiological signal-based HCIs have also found applications in non-strictly related medical fields, such as emotions recognition [9], smart home control [10], drivers' distraction avoidance [11] and musical expression [12].

Independently from the input signal, a unique functional model is accepted to describe HCI systems, see figure 1 [13], made by the following components: (1) the *acquisition,* which concerns the signal measurement and data transmission; (2) the *transducer,* which is devoted to the extraction of the *features*, that are special characteristics of the measured signals, and to the recognition of the user's intent, typically by means of a classification algorithm; (3) the *control interface,* which translates the classification output into a control command for the external device; (4) the *output device*, which is the peripheral to be driven. A visual, acoustic and/or tactile feedback is provided from the device back to the user in order to allow performance adjustments.

The core of the whole HCI chain is the detection of the patterns associated to the user's volition: while the user must be trained at correctly performing the task, the *classifier*, which is a set of software routines, must be trained to correctly recognize the particular task among a set of others, which are the *classes*. For these reasons, many studies in the literature have been devoted to find classification algorithms with the accuracy as high as possible (see [14] for a review of classification techniques in EEG-based BCIs and [15] for details about EMG classification). Among classifiers, support vector machines (SVMs) have been widely implemented for HCI due to their versatility and robustness with non-stationary data [16]. Moreover, SVMs can be easy to implement even for non-experts, thanks to the availability of different free toolboxes for SVM-based classification, e.g. LIBSVM [17], SVM-light (http://svmlight.joachims.org/), SVMTorch [18], mySVM (www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html), just to name a few, or to the many SVM implementations in Matlab (The MathWorks Inc., Natick, MA, 2000). However, if not fully understood, sometimes a non-optimal choice of the SVM parameters for the classification can be adopted. Moreover, many works lack in details on the setting of SVM, which are strategical in replicating the analysis and the results, as it occurs for research matters.

The objective of this review is to describe the use of SVMs in the HCI field, with the aim to provide some practical hints for a correct SVM-based classification. We limited the search to electrophysiological signal-based HCIs, in particular EEG and EMG-based, and mainly to HCIs used for device control. A description of the main theoretical aspects underlying SVMs is provided, including an overview of the mostly adopted SVM implementations, and details are tabled. A statistical analysis of the occurrences of SVM features in the literature is performed, including a critical comparison with other classifiers.

Source bibliography comes from the main online databases, in particular Pubmed, Scopus, IEEE *Xplore,* up to year 2015. The list of all the main acronyms used in the manuscript can be found in appendix.

## 2. Introduction to SVM-based classification

For the sake of clarity, the following introduction to SVM will start from intuitive geometric concepts. Nevertheless, SVM classification has very strong theoretical bases in the theory of statistical learning developed by Vapnik and Chervonenkis [19].

### *2.1. Margin maximization*

In machine learning, a classification problem consists in the identification, within a set of categories, of the category a new *observation* belongs to; such identification is performed on the basis of the information previously deducted from a set of observations whose category membership is known. The phase of information extraction is called *training*, while the phase of unknown instances categorization is called *testing*.

Let's consider the binary classification problem depicted in figure 2, where squares denote objects belonging to class 1, and circles represent objects belonging to class 2. In principle, the separation in two classes can be realized by any line (in general infinite lines) that separates the two regions containing only squares and only circles, respectively, as the examples line 'A' or line 'B' in figure 2. Intuitively, line 'A' seems to achieve a better separation between classes with respect to line 'B', since it separates with a *safer margin*, key concept of the SVM approach towards classification [20].

In the binary case, that is, the classification between only two classes, in a multi-dimensional space, SVM is used to find the hyperplane having the maximum distance (or *margin*) from both classes [21]. Regardless the class, the points closest

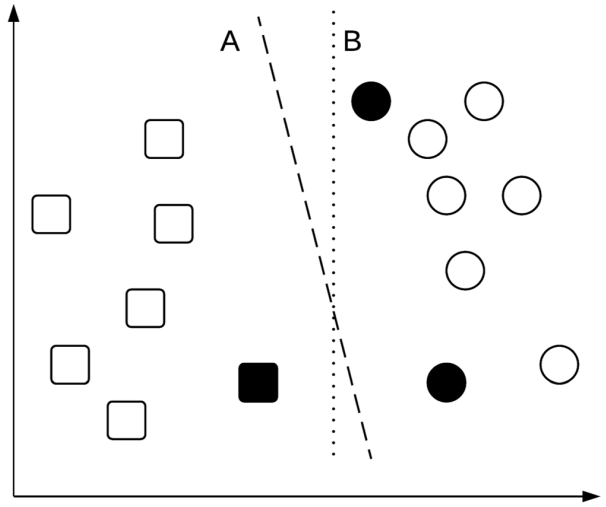


**Figure 2.** A binary classification problem.

to the hyperplane are called *support vectors* (black squares and circles in figure 2).

Let $x_i \in R^n$, with $i = 1, 2, \ldots N$, be the $i$th point of a set $S$, $n$ being the total number of features and $R$ the space of the features; $x_i$ can belong to class $\omega_1$ or to class $\omega_2$, which are assumed to be linearly separable. A hyperplane in $R^n$ can be written as $w^t x + w_0 = 0$, where $w \in R^n$ is an $n$-dimensional *weight vector*, and $w_0$ is a *bias term*. Many conventional hyperplane-based classifiers, e.g. linear discriminant analysis (LDA) [22, 23], aim at finding optimal values for $w$ and $w_0$ so that $w^t x_i + w_0 > 0$ if $x_i$ belongs to class $\omega_1$, and $w^t x_i + w_0 < 0$ if $x_i$ belongs to class $\omega_2$ (the case $w^t x_i + w_0 = 0$ is a point of uncertainty and $x_i$ is typically assigned to one of the two classes arbitrarily). Differently, SVM does not only require that the training patterns lie on the correct side of the decision boundary, but also requires the *safety margin*, for a better generalization capability. In particular, during the training phase we require more stringent inequalities, such as:

$$w^t x_i + w_0 \geqslant 1 \quad \text{if } x_i \text{ belongs to class } \omega_1 \tag{1}$$

$$w^t x_i + w_0 \leqslant -1 \text{ if } x_i \text{ belongs to class } \omega_2 \tag{2}$$

Since such relations can be trivially satisfied by taking a large enough $w$, the maximization of the margin is obtained with the minimization of the norm of $w$, $J(w) = \frac{1}{2} w \cdot w$, bounded by the above constraints. $J(w)$ is called the *objective function* (1/2 is for computational convenience).

Finally, in the test phase, we classify the new instances according to the usual rules:

$$\text{If } w^t x_i + w_0 > 0\,, \text{ then assign } x_i \text{ to class } \omega_1 \tag{3}$$

$$\text{If } w^t x_i + w_0 < 0\,, \text{ then assign } x_i \text{ to class } \omega_2 \tag{4}$$

It turns out that the above described optimization problem can be put in terms of a *convex quadratic program* and therefore: (i) it has a single global optimum and (ii) it can be solved using well-known techniques (see [24] for further details). In addition, this program has a number of relevant properties:

- Its complexity depends on the number of training instances only, i.e. the size of the training dataset, and not on the feature space dimensionality. This is a very important peculiarity of SVMs, which makes them insensitive to the so-called 'curse of dimensionality', a major concern when designing EMG and EEG-based systems [14]. The 'curse of dimensionality' depends on the fact that, if the number of training data is small compared to the number of extracted features, the classifier will probably perform poorly due to insufficient data to build the classification rule. This curse affects mainly BCI systems as small training samples are usually available (training is consuming for the subjects) and many channels (and features) are needed to describe the classification problem.
- The weight vector $w$ depends only on the training patterns that lie on the margin (for those instances we have $w^t x_i + w_0 = \pm 1$), i.e. the support vectors.
- Given $w$, it is simple and straightforward to compute the bias parameter $w_0$.

### 2.2. Non linearly separable data: the soft-margin

When data are not linearly separable, no hyperplane that perfectly discriminates classes exists. Consequently, we can find a hyperplane with the lowest error, as our best. In such an occurrence, two error sources are considered: *misclassifications*, i.e. points that lie on the wrong side of the hyperplane, and *within-the-margin anomalies*, i.e. points that lie on the correct side of the hyperplane but within the margin. To model those errors, a slack variable $\xi_i \geqslant 0$ is introduced for each training instance $x_i$. If $x_i$ is correctly classified, then $\xi_i = 0$. If $x_i$ lies within-the-margin or gets misclassified, then $\xi_i$ is set to the distance of $x_i$ from the separating hyperplane. In this way, the constraints of the optimization problem become:

$$w^t x_i + w_0 \geqslant 1 - \xi_i \text{ if } x_i \text{ belongs to class } \omega_1 \tag{5}$$

$$w^t x_i + w_0 \leqslant -1 + \xi_i \text{ if } x_i \text{ belongs to class } \omega_2 \tag{6}$$

and the sum of the slack variables, i.e. the overall error, is added as a *penalty factor* to the objective function. The resulting program can then be solved similarly to the linearly-separable data case [24].

Usually, a regularization parameter $C$ is introduced to weight the penalty term in the objective function, which then becomes $J(w, \xi) = \frac{1}{2} w \cdot w + C\left(\sum_{i=1}^{N} \xi_i\right)$, thus allowing the experimenter to trade the training set accuracy off for the expected generalization capability. If a large value of $C$ is chosen, then the resulting hyperplane will commit fewer errors on training data but will be characterized by a smaller margin (thus a minor expected generalization capability). On the contrary, a small value of $C$ will lead to an SVM with greater expected generalization capability (larger margin) but misclassifying more training instances. The soft-margin implementation is advisable for EEG and EMG classification: in fact, both signals are often characterized by high levels of outliers and noisy examples, which can derive from artefacts (e.g. motion artefacts, equipment artefacts, etc.) and by a poor

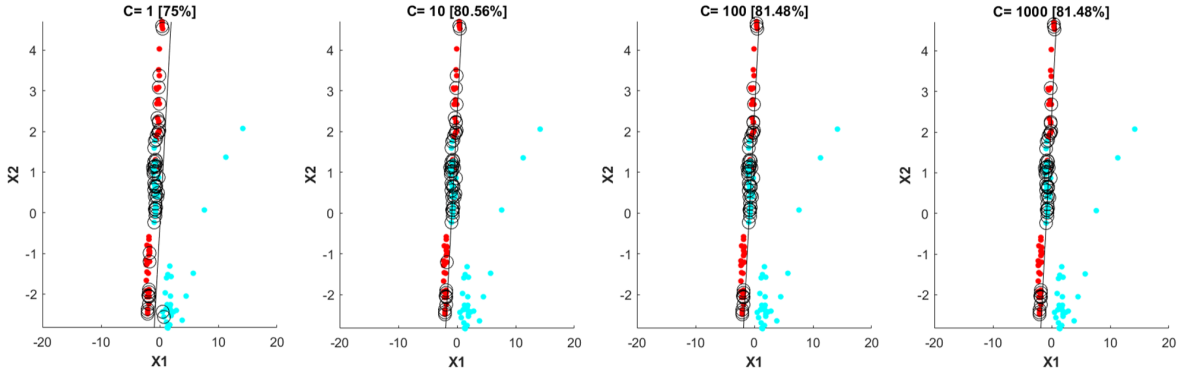


**Figure 3.** Binary classification with a linear SVM and different values of $C$ (1, 10, 100 and 1000). Red dots represent samples in class 1, cyan dots represent samples in class 2, and circles indicate support vectors. X1 and X2 represent the first two dimensions of the feature vector. Accuracy is reported in brackets. Data come from an EMG-based protocol, where the subject was asked to perform 5 different hand gestures (see [25] for details on protocol implementation and data analysis). For the sake of simplicity, just two tasks are considered.

signal-to-noise ratio. The possibility to have adjustable margins which take into account the effect of outliers in the training dataset is definitively beneficial.

The choice of $C$ is critical, leading to *overfit* or *underfit* risks for too high or too low values of $C$ respectively, as schematically represented in figure 3. A binary classification problem is depicted. It can be easily seen how the choice of different values of $C$ affects the number of support vectors and, definitively, classification performances, which range from 75% with $C = 1$ to 81.48% with both $C = 100$ and $C = 1000$.

No optimal criteria are given to set a value for $C$, but grid-search and cross-validation can be considered, assigning values within $10^{-6}$–$10^{+6}$ on a logarithmic scale [24]. The introduction of the slack variables simply softens the aforementioned margin, so that we can refer to *soft margin SVM* or $C$-SVM; it represents the standard configuration for a typical SVM classification problem.

Even though the value of $C$ can be associated to the extent of the margin, such a relationship is difficult to visualize and quantify. Therefore, it is worth considering another implementation of the soft margin concept, called $\nu$-SVM, which results in an easier interpretation, thanks to the introduction of the parameters $\nu$ and $\rho$ [26], with $\nu \in [0, 1]$ and $\rho \geqslant 0$. The penalty factor becomes:

$$-\nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i \tag{7}$$

and the constraints:

$$w^t x_i + w_0 \geqslant \rho - \xi_i \text{ if } x_i \text{ belongs to class } \omega_1 \tag{8}$$

$$w^t x_i + w_0 \leqslant -\rho + \xi_i \text{ if } x_i \text{ belongs to class } \omega_2 \tag{9}$$

The role of $\nu$ and $\rho$ can be figured out considering that when all instances lie on the correct side of the hyperplane, and outside the margin, $\xi_i$ equals zero for all input data $x_i$, and the constraints (8) and (9) reduce to $w^t x_i + w_0 \geqslant \rho$ and $w^t x_i + w_0 \leqslant -\rho$. It follows that $2\rho/w$ is the margin that separates the classes. In general, considering misclassifications and within-the-margin anomalies, $\rho$ is linearly related to the size of margin. In the occurrence of $\rho = 1$, corresponding to the soft margin constraints, according to the equation (7), the term $(1/N)\sum_{i=1}^{N}\xi_i$ represents the fraction of errors relative to the training data, while $\nu$ represents a sort of compensation: the higher the value of $\nu$, the lower the penalty factor. In other words, $\nu$ indicates the fraction of errors we can accept. It turns out that this interpretation is valid for any $\rho \geqslant 0$ and that $\nu$ represents both an upper bound on the fraction of errors (misclassifications and within-the-margin anomalies), and a lower bound on the fraction of input instances that will be selected as support vectors (see [27] for further details).

All considered, the $\nu$-SVM allows a more evident and direct interpretation of the parameters with respect to $C$-SVM, but needs the optimization of two variables ($\nu$ and $\rho$) instead of one ($C$).

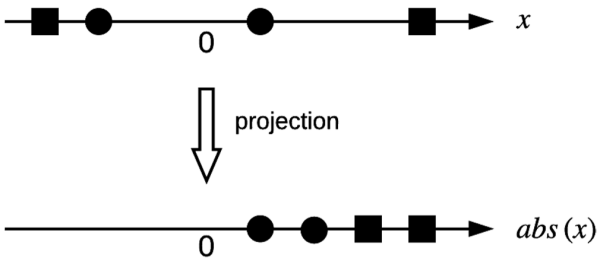


**Figure 4.** Space Projection. The two classes, circles and rectangles, are non-linearly separable in the original space (the top arrow), but become linearly separable in another space provided by the *abs*(-) function (the bottom arrow).

### 2.3. Non linearly separable data: the kernel trick

*Soft-margin* admits a certain error so to allow a linear approach for non-linearly separable data. Such an approach is useless in case the separating function is 'inherently' non-linear, so that it is worth to consider the option of *projection*. In fact, data not-linearly separable in their original space can get linearity when mapped into another one (especially when of higher dimensionality), so that SVM can be further applied.

As an example, figure 4 shows four mono-dimensional instances belonging to two classes (circles versus rectangles).

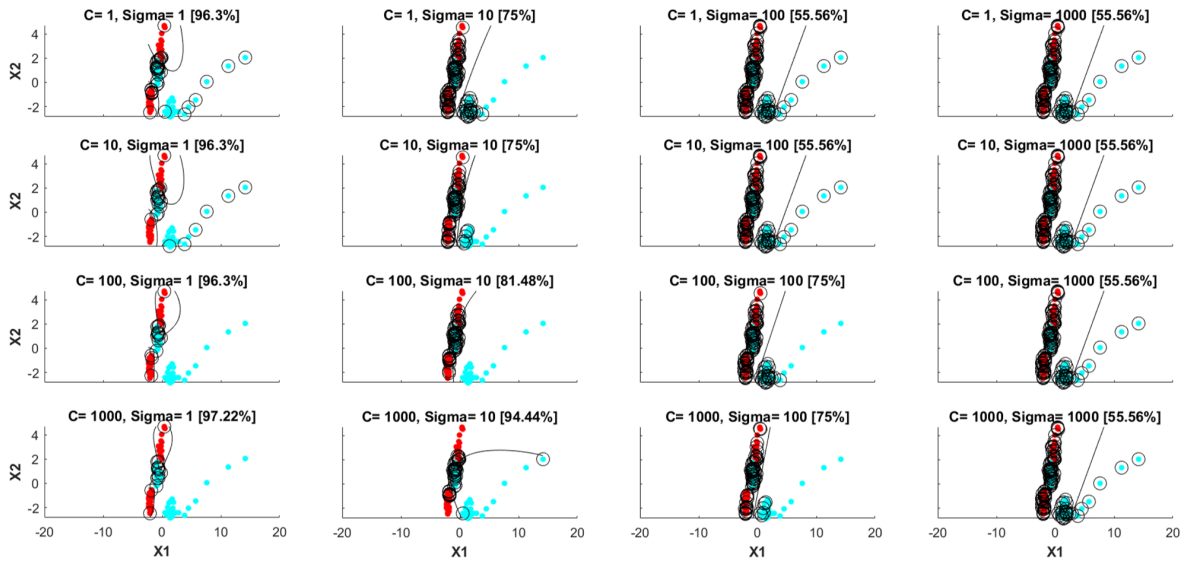


**Figure 5.** Binary classification of EMG data with an SVM with RBF kernel. $C$ and $\sigma$ can be 1, 10, 100 and 1000. Red dots represent samples in class 1, cyan dots represent samples in class 2, and circles indicate support vectors. X1 and X2 represent the first two dimensions of the feature vector. Accuracy is reported in brackets. Data come from an EMG-based protocol, where the subject was asked to perform 5 different hand gestures (see [25] for details on protocol implementation and data analysis). For the sake of simplicity, only two tasks are considered.

In the original space (top arrow), no single point separates the two classes, but the *abs* function provides mapping into another space allowing linear separation between classes (bottom arrow).

Let $\phi : R^n \rightarrow H$ denote the mapping function. Firstly, each input instance $x_i$ is mapped according to $\phi(x_i)$. Then we train our SVM model, typically using a soft-margin formulation, to discriminate the mapped points. Adopting the quadratic program computations [24], the training algorithm will depend on $\phi$ only trough dot products of mapped instances, i.e. $\phi(x_i) \cdot \phi(x_j)$. Since there must exist a function $K : R^n \times R^n \rightarrow R$ such that $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, it is sufficient to use $K$ in the training algorithm, avoiding computing the mapping functions $\phi(x_i)$ and $\phi(x_j)$ explicitly. $K$ is known as the *kernel function*, while the whole procedure is known as the *kernel trick*, as it allows advantages of the mapping procedure without higher computational costs. The fact that the decision rule of SVM is a simple linear function in the kernel space makes SVM stable and characterized by low variance (variance reflects the sensitivity of the classifier to the training set used [14]). This is a useful property when dealing with EEG and EMG data, which are non-stationary signals with features changing over time; low-variance classifiers, such as SVM, can cope with such signals better than others.

A valid kernel function has to be positive definite (Mercer's condition), symmetric, and has to reflect the similarity between its inputs, i.e. $K(x_i, x_j)$ should be high (low) if $x_i$ and $x_j$ are similar (dissimilar) with respect to the problem at hand [20]. The most used kernels are:

- Radial Basis Function (RBF), $K(x_i, x_j) = \mathrm{e}^{-\frac{\|x_j - x_j\|^2}{2\sigma^2}}, \sigma \neq 0.$
- Polynomial, $K(x_i, x_j) = (x_i \cdot x_j + 1)^d, d > 0.$
- Sigmoidal, $K(x_i, x_j) = \tanh(kx_i \cdot x_j - \delta).$
- Cauchy, $K(x_i, x_j) = \left(1 + \frac{\|x - y\|^2}{2\sigma^2}\right)^{-1}, \sigma \neq 0.$
- Logarithmic, $K(x_i, x_j) = -\log(\|x - y\|^d + c), d > 0.$

The linear problem can be considered as a subset of the non-linear one with kernel $K(x_i, x_j) = x_i \cdot x_j$.

For each kernel choice, the values of other parameters, besides $C$, should be set. For example in the popular RBF kernel the experimenter has to choose also the kernel width $\sigma$. To do so, the *grid-search* is usually adopted; anyway other techniques, such as particle swarm optimization (PSO) [28] and genetic algorithms (GA) [29] are also used in the literature.

Values determined for the kernel and the related parameters greatly influence the decision surface and, consequently, the classification performance. In figure 5 and for the same dataset used for figure 3, an RBF kernel is optimized considering, for both $C$ and $\sigma$, values of 1, 10, 100 and 1000. According to the results, the selection of $C = 1000$ and $\sigma = 1$ allows an accuracy as high as 97.22%. In figure 6, instead, the results of the optimization of a polynomial kernel-SVM with degree $d$ are reported, considering for $C$ values of 1, 10, 100 and 1000 and for $d$ values of 2, 5, 7 and 10. As a result, the highest accuracy (97.22%) is obtained both with $C = 1000$, $d = 2$ and with $C = 1000$, $d = 7$. In case of equal accuracies, the time spent for the training can be used as a metric to choose the most performing configuration.

### 2.4. Multiclass SVM

In principle, the margin criterion of SVM can perform multiclass classification but it is useless, resulting in a quadratic program with too many variables to be optimized. Differently, more computationally efficient techniques, although potentially inaccurate, are typically used in order to build a

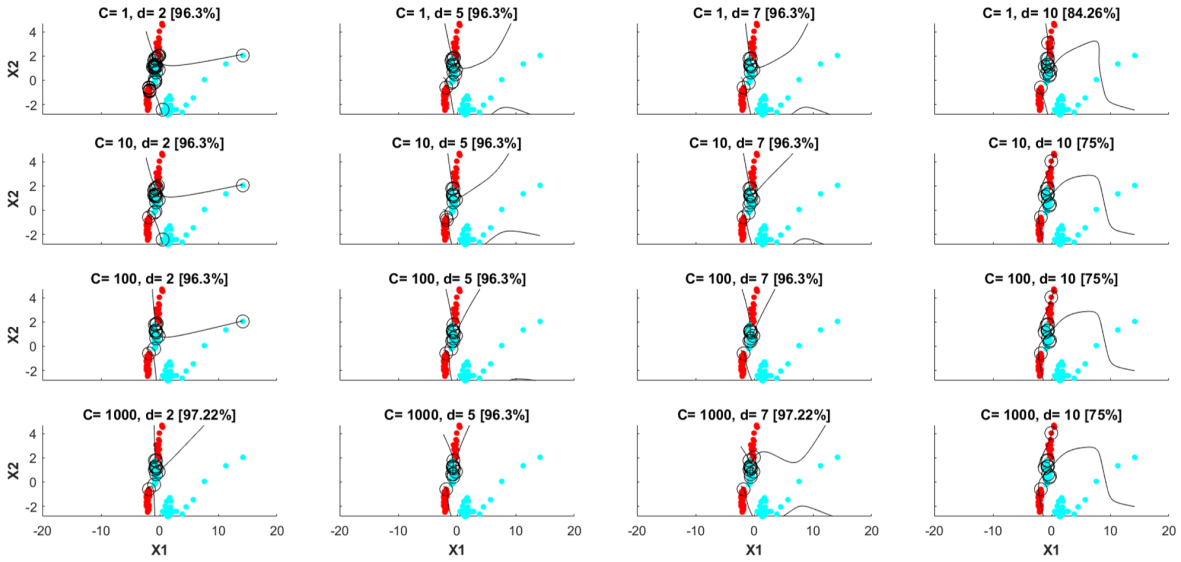


**Figure 6.** Binary classification of EMG data with an SVM with polynomial kernel. $C$ can be 1, 10, 100 and 1000, while the degree ($d$) of the polynomial can be 2, 5, 7 and 10. Red dots represent samples in class 1, cyan dots represent samples in class 2, and circles indicate support vectors. X1 and X2 represent the first two dimensions of the feature vector. Accuracy is reported in brackets. Data come from an EMG-based protocol, where the subject was asked to perform 5 different hand gestures (see [25] for details on protocol implementation and data analysis). For the sake of simplicity, only two tasks are considered.

*multi-class* SVM starting from many 2-class SVMs [22, 24], namely the One-Vs-One (OVO) (also called in the literature One-Against-One (OAO)) and the One-Vs-All (OVA) (also called in the literature One-Against-All (OAO) and One-Against-Rest (OAR)).

For OVO, the $M$-class problem is split into $M(M-1)/2$ binary problems aimed at separating one class from another. To obtain the final response, i.e. the class to which a new input instance is expected to belong, a majority voting strategy is employed.

For OVA, $M$ classifiers are built. The task of the $i$th ($i = 1, \ldots, M$) classifier is to separate the instances belonging to class $i$ from all the others. When a new input has to be classified, every classifier is asked for its corresponding score and the class having the highest result is selected as the final response.

All those strategies can have the drawbacks to determine unclassifiable/uncertain regions in the feature space, and to meaningfully increase the computational time with the number of classes. In order to overcome such limitations, the directed acyclic graphs (DAG) approach, consisting in assigning unclassifiable regions to the classes associated with the leaf nodes of a decision tree, was suggested [30].

### 2.5. SVM variants

In view of the intrinsic limitations of each of the aforementioned approaches, many variants of SVM have been developed; the most relevant for our purposes are:

- *Least-squares SVM (LS-SVM)*: differently from solving a quadratic programming problem with linear inequality constraints with standard SVM, LS-SVM involves solving a set of linear equations, thus making the solution less time-consuming in the presence of large training sets [31]. Large datasets are typical of EMG-based studies, where a huge number of classes is usually taken into account (e.g. 18 different hand gestures in [32]) and to a minor extent of BCI protocols which usually involve less classes (2–5).
- *Fuzzy SVM (FSVM)*: allows integrating in SVM a means for assigning a *utility* value to the training data [33]. Indeed, in many real-world applications, some of the training instances could be more important than the others, and the learning algorithm should take into account this difference. This is useful when the training set is suspected of containing outliers and/or mislabeled points, a condition typical of both EEG and EMG data which are highly sensible to different sources of noise.
- *Fuzzy least-squares SVM (FLS-SVM)*: overcomes the issue of unclassifiable regions in the feature space, which can occur with standard SVM or LS-SVM in solving multiclass problems by means of the OVO or the OVA approach [34]. This advantage can be obtained using learner-specific fuzzy membership functions, combinable to obtain (fuzzy) class-specific membership functions that are well-defined in each portion of the feature space. This implementation can be very useful when dealing with EMG-based datasets, which usually contain a huge number of classes.
- *Multiple kernel learning SVM (MKL-SVM)*: the kernel function becomes a linear or nonlinear combination of multiple base kernels [35]. Consequently, data can come from non-homogeneous information sources. Moreover, the optimal combination is itself learned from data, thereby eliminating the need for a preliminary, possibly

arbitrary, kernel selection. This can be very useful for reducing variance in the problem and making SVMs more stable, a property that allows them to cope with non-stationary data such as EEG and EMG.

- *Spatially-weighted SVM (sw-SVM)*: proposes to integrate spatial feature weights into the SVM optimization problem and tune these weights as if they were hyperparameters [36]. In this way the classifier learns from spatially filtered data, thus improving class separation, while reducing errors. This implementation allows to enhance just the weight of the most informative channels and can be efficient in case of high-density datasets, typical of EEG recordings.
- *K-means SVM*: a *k*-means procedure can be implemented before SVM classification in order to avoid possible shortcomings due to input data redundancy and to speed up training [37]. Clusters containing only vectors belonging to the same class can be disregarded, while the others are retained and considered. SVM training is then carried on the new reduced training dataset, with a noteworthy reduction of complexity. This could be a valuable property when dealing with the classification of large datasets, such as the EEG and EMG ones.
- *Relevance vector machine (RVM)*: it can overcome some of the typical limitations of SVMs (e.g. determination of the $C$ parameter, non-probabilistic predictions) [38]. RVMs have the same functional form of SVMs but within a Bayesian framework.
- *Twin SVM*: uses two different SVMs, one for each class, for binary problems, and generates two distinct hyperplanes, each one being close to the patterns of the relative class, '1' outdistanced from the point of the other class. The class of a point $x$ is determined by the closest hyperplane. Twin SVM allows a smaller number of constraints with respect to a standard SVM, and enhances operational speed [39], a property valuable for online implementations of HCI systems.
- *$\varepsilon$-support vector regression ($\varepsilon$-SVR)*: SVMs can be used also for regression [40]. The task of $\varepsilon$-SVR is to find a continuous function $f(x)$ that has at most $\varepsilon$ deviation from the actual targets of data samples and is as flat as possible. This is equivalent to estimating regression coefficients of $f(x)$ with these requirements. It can happen that the linear function $f$ is not able to fit the training data. Hence, as for the classification case, slack variables $\xi$ (and therefore $C$) or kernels can be introduced.

### 2.6. Performance evaluation

The performances of SVMs, and of classifiers in general, are usually evaluated by means of *accuracy* (or *success rate* or *recognition rate*), defined as the number of correct classifications over the total classifications. They can also be expressed in terms of *error rate* (or *misclassification rate*), defined as (100 − accuracy)% and representing the number of errors over the total classifications. If a classifier is not able at distinguishing the classes, its performances equate the *chance level* (50% of accuracy in a binary problem).

Different methods can be used to evaluate accuracy:

- *Train-Test* evaluation: consists in randomly or sequentially splitting the whole dataset into a training and a testing set and in computing accuracy on the testing set.
- *N-fold cross validation (CV)*: consists in partitioning the whole dataset in $N$ not-intersecting subsets, approximately of the same size, and in using the $i$th subset as a testing set and the remaining $N-1$ subsets as training set; the process iterates until each subset is used once as a testing set, and the accuracy comes from the average of the $N$-tests accuracies.
- *Leave-one-out (LOO) CV:* consists in leaving one sample out of the whole available dataset to build the testing set and using all the remaining samples as training set; the process iterates until each sample of the dataset is used once as a testing set.

*CV* is more computationally demanding than *train-test*, but allows to have an unbiased performance estimate, since all data are used (at least once) for testing; in this way the performance is independent from the particular dataset division which is adopted.

When accuracies of different classifiers need to be compared and statistical significance among different classifiers needs to be assessed, *confidence intervals* and *statistical tests* can be used [41]. Although these methods are well-founded and can provide the experiment with more reliable estimates, they are rarely used in HCI studies because of the huge amount of data necessary for meaningful results.

Even if widely used, accuracy has some limitations as an evaluation criterion, as it does not take into account the class distribution among examples (e.g. less frequent classes have smaller weight in the total accuracy) and the loss of information due to different types of errors (e.g. did the classifier misclassify the class or simply abstain from classification? [42]). For these reasons, many indicators can be introduced in order to have a more realistic evaluation of the whole classification process (e.g. sensitivity, specificity, precision, recall, Cohen's kappa coefficient, Area under Roc Curve (AUC), etc.). The interested reader can refer to [43] for a generic overview of evaluation criteria in machine learning and to [44] for an overview of performance evaluation in BCI and HCI in general.

## 3. EEG-based HCI and SVM

### 3.1. Overview of BCI systems

BCIs rely on the recording of user-modulated brain signals to drive a device. Brain signals can be measured by means of different technologies either invasive, such as the ElectroCorticoGraphy (ECoG) [45] or non-invasive, such as the EEG [46], the functional Near-Infrared Spectroscopy (fNIRS) [47], the MagnetoEncephaloGraphy (MEG) [48], the functional Magnetic Resonance Imaging (fMRI) [49], etc.

In this paper we focus on EEG-based BCIs, since they are popular for device control, thanks to the manageability and relatively low cost of the EEG technique. Anyway the interested reader can refer to the many reviews devoted to BCI

(e.g. [50] for a complete review of processing algorithms for fNIRS-BCIs or [51] for a review of ECoG-based BCIs) to have a deeper understanding of the other techniques.

Many brain patterns have been used and many protocols have been implemented for BCI. The most studied ones are based on event-related potentials (ERPs), which consist in brain responses evoked by external stimulations when the user performs a specific mental activity. This activity could be, for example, the mental discrimination of a rare acoustic stimulus within a set of frequent stimuli, the discrimination of a verbal incongruity or of a known face among anonymous ones, etc. These cognitive tasks evoke in the EEG a specific ERP response, which can be discriminated by means of algorithms and used to control a device. The most relevant application of ERP-based BCIs is the P300-speller [52, 53]. A subject is asked to focus the attention on a particular symbol in a matrix of symbols whose rows and columns iteratively flash. Three hundred milliseconds after the flashing of the row/column containing of the chosen symbol, a positive voltage spike, named P300, is measured on the subject's scalp. When the P300 occurs, the classifier recognizes the symbol, so allowing the subject to communicate. Other than a symbol, ERP-based BCIs were used to recognize an intention of movement to drive a wheelchair [54], to drive an analogue mouse [55], a mobile robot [56], a domotic interface [57], to surf on the internet [58], to paint [59], to browse photographs [60], etc.

The control of a device can be also achieved by means of sensorimotor rhythms (SMRs)-based BCIs, on the basis of brain signal variations evoked by actual or imagined mental tasks [61]. When a subject moves or imagines to move a body segment or to perform a specific mental task, alpha (8–13 Hz), beta (14–26 Hz) and gamma (>30 Hz) frequency bands, recorded on the brain sensorimotor cortex, vary in voltage amplitude (the first two increasing and the third decreasing). After a training stage, the subject learns how to voluntarily modulate those voltages and thus control a device, e.g. a helicopter in a 3D space [62], a cursor on a screen [46], a robotic arm [63], a wheelchair [64], a spelling program [65], etc.

The third big class of BCIs is based on steady-state visually evoked potentials (SSVEP), which are responses evoked in the EEG spectrum by long trains of flickering visual stimuli and characterized by a frequency peak at the same frequency of the stimuli themselves. Being easily recognizable in the EEG spectrum, SSVEPs have been widely used to implement BCIs, with high accuracy and throughput rates, devoted e.g. to the selection of buttons on a screen [66], to the communication by means of a speller [67], to the driving of a wheelchair [68], to the control of a robotic arm [69], etc.

Finally, there is a class of BCI systems that make use of slow-cortical potentials (SCPs), which are slow (<1 Hz) negative or positive potential shifts voluntary modulated by a subject, to implement a binary communication [70]. SCPs-based BCIs have not found a wide diffusion for communication and control purposes, due to the slowness of brain responses, but were exploited for implementing neuro-rehabilitation strategies [71].

The following sections concern the review of the BCI literature, focusing on the exploited brain features: ERPs, SMRs, SSVEPs and SCPs. Details of the SVM implementation and its performances are reported, making a comparison with those of other classifiers. The BCI protocol, the system and the applications are run over or just mentioned for the sake of simplicity.

### 3.2. Event-related potentials (ERPs)-based BCI

For typical speller-based applications, SVM was successfully adopted in [72], where a self-training semi-supervised linear-kernel SVM achieved an accuracy similar to that of a standard SVM (up to 98.5%) by using a smaller training dataset. In [73] an ensemble of linear SVMs achieved the same accuracy of a standard SVM, 96.5%, with the full training set, and accuracy higher by 5% with 1/3 of the training set. In [74] a sequential updating self-training LS-SVM, whose kernel gradually improved with the insertion in the training set of upcoming unlabeled data, was used with an online spelling accuracy greater than 85%. In [75] data from a P300 speller, with modifications in symbol size, inter-symbol distances and speller colors, were classified with an RBF-SVM and an LDA, with RBF-SVM performing better in each condition (in the best configuration, accuracy up to 90% was achieved by SVM versus 80% of LDA).

In ERPs-based protocols different from the speller one, as the one in [36], spatio-temporal filtering and an ensemble of linear sw-SVMs were used to classify data acquired from a visual feedback experiment, consisting in memorizing the position of a set of digits and in indicating the exact position of a random target number. The obtained classification accuracy was 87.80% $\pm$ 3.63%, higher than the ones achieved by linear SVM and simple sw-SVM (70.71% $\pm$ 10.77% and 80.71% $\pm$ 6.61%, respectively). In [76] authors compared an RBF-SVM against a linear SVM and a linear logistic classifier (LLC) in classifying ERPs acquired during an image rapid serial visual presentation (RSVP) protocol. SVMs outperformed LLC, while RBF-SVM was more accurate than linear-SVM. In [77] ERPs following true and false statements were classified with the aim to separate covert 'yes' from covert 'no'-thinking. Four classifiers were compared, namely a linear SVM, an RBF-SVM, a stepwise LDA and a shrinkage LDA. All the classifiers performed at chance level when separating 'yes' from 'no'; however, when the single responses were discriminated against baseline, RBF-SVM showed the highest accuracy (68.8%). In [9] a non-verbal communication-based BCI was created on the basis of the ERP associated to implicit negative emotional responses to specific neutral faces. SVMs were used with both linear and RBF kernels, with features both in time and time/frequency domains. The classifiers exhibited accuracy up to 80% in discriminating emotional responses. In [78] the imagination of Japanese vowels vocalization was investigated as an input to control a speech prosthesis. An RBF-SVM, a linear RVM (RVM-L) and an RVM with Gaussian kernel (RVM-G) were compared. Accuracy obtained by RVM-L was around chance level and also significantly worse than RVM-G's one. Linear classification was ineffective for silent speech. In comparison to SVM-G, RVM-G achieved slightly better accuracy but with a significant reduction of relevant vectors, while its accuracy worsened in case of few training points.

**Table 1.** SVM features in ERPs-based BCI papers. AUC = area under ROC curve. NA = not available.

| Paper | Protocol | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [72] | Speller | 98.5% | Self-training semi-supervised | Linear | 2 | — | Fisher score | Self-training semi-supervised | LIBSVM |
| [73] | Speller | 96.5% | Ensemble SVM | Linear | 2 | — | Five values of C | Train-test | NA |
| [74] | Speller | 85% | LS-SVM | Linear | 2 | | Fixed | Train-test | LIBSVM |
| [75] | Speller | 90% | Standard | RBF | 2 | — | 5-fold CV | 3-fold CV | LIBSVM |
| [36] | Visual feedback | 70.71% ± 10.77%<br>80.71% ± 6.61%<br>87.80% ± 3.63% | Standard sw-SVM<br>Modified sw-SVM | Linear | 2 | — | Fixed | 5-fold CV | NA |
| [76] | RVSP | AUC = 0.927<br>AUC = 0.941 | Standard | Linear<br>RBF | 2<br>2 | — | Fixed<br>10-fold CV | Monte Carlo<br>Monte Carlo | NA |
| [77] | Yes/No discrimination | 68.8% | Standard | Linear | 2 | — | 10-fold CV | 10-fold CV | LIBSVM |
| | | | | RBF | 2 | — | 10-fold CV | 10-fold CV | |
| [9] | Emotion recognition | 80% | $\nu$-SVM | Linear | 3 | NA | $\nu$ Fixed | 36-fold CV | LIBSVM |
| | | | | RBF | | | $\gamma$ different values | | |
| [78] | Letters<br>Imagination | 77%<br>50%<br>79% | Standard<br>RVM<br>RVM | RBF<br>Linear<br>RBF | 2 | — | $C$ Fixed, $\sigma$ CV | Train-test | SVM and Kernel Methods Matlab toolbox; Sparse Logistic Regression toolbox |

Table 1 summarizes the main SVM features used in ERPs-based BCI works, namely type of protocol, maximum achieved performances, SVM implementation, type of kernel, number of classes involved in the classification, type of multiclass implementation, adopted methodology to set hyperparameters, chosen methodology to evaluate performances, used tool. In particular, performances refer to the best obtained result, in terms of accuracy, or error rate, or AUC, etc. If not differently indicated, performances correspond to accuracy. In any case, a comparison among performances of different systems cannot be directly evidenced since the different implementations used.

Table 1 evidences that SVM, either with linear or RBF kernel, achieves high accuracy in ERPs discrimination and near 100% accuracy in speller applications. The implementations different from the standard $C$-SVM (sw-SVM for example) can boost ERPs-based BCI system performances, accordingly to the type of data and the final application. Hyperparameters setting is performed in different ways, mainly by fixing their values or by using CV. CV remains the method of choice for performance evaluation. As a final remark, LIBSVM results to be the most used toolbox.

### 3.3. Sensor-motor rhythms (SMRs)-based BCI

The SMRs-based BCI review regards papers dealing with different mental tasks which can be pure motor tasks, both motor and non-motor tasks and other typologies of tasks.

*3.3.1. Motor tasks.* The imagination of left and right hand movements is widely used in BCI, because such tasks evoke highly discriminable patterns in well-defined and opposite sides of the brain. For these reasons, many studies have tried to find the best combination of extracted features and classifier to boost their detection performances. For example, spatial and temporal principal component analysis (PCA) and a linear SVM were used in [79] with accuracy of 73.65%, while in [80] an RBF-SVM outperformed five different classifiers (Linear Mahalanobis Distance, Quadratic Mahalanobis Distance, Bayesian Classification, two types of artificial neural networks (ANN)) independently from the spatial and the temporal filter used for the preprocessing. Also in [81] three different feature-extraction methods and three different classifiers (LDA, Adaboost and RBF-SVM) were compared, with SVM and LDA performing better (error rates of 0%–23.81% and 9.61%–14.33%, with SVM and LDA respectively) both with single features and with combinations of features. In [82] RBF-SVM outperformed LDA and ANN both with the proposed features (9% versus 10% of ANN and 12% of LDA) and with simple AR features (18% versus 21% of ANN and 22% of LDA). In [83] an RBF-SVM and a fuzzy RBF-SVM were compared with ANN and LDA and outperformed them in terms of amount of information transmitted per second. In [84] an RBF-SVM was used in combination with neuro-fuzzy prediction and multi-resolution fractal feature vector, with accuracy of 91%. In [85] authors proposed a kernel fisher (KF)-posterior-probability (PP)-SVM, consisting in calculating the within-class scatter matrix of the two classes, in integrating it into the RBF kernel of an LS-SVM, and in calculating the output of the classification with posterior probabilities. When compared to a simple SVM, a PP-SVM and a KF-SVM, the KF-PP-SVM achieved higher accuracy (75.73% versus 70.86%, 71.52% and 74.11%). Finally, in [86] a feature selection technique, based on a genetic algorithm (GA) and an RBF-SVM, resulted with accuracy higher than

that obtained with feature combination (80.8% versus 72.3%). With the adoption of GA, SVM showed higher accuracy with respect to ANN (80.8% versus 75.6%).

In some studies, the imagery of feet movements was used together with other mental tasks, as in [87] where a linear SVM was compared to LDA, Mahalanobis Distance, Generalized Distance Based classifier and Bayes classifier, in the discrimination of left hand, right hand, feet and tongue movements. Accuracy higher than 80% was achieved by both LDA and SVM. Similarly, right foot imagery was classified in [88], where authors compared a Bayesian LDA (BLDA) with two different versions of LDA (simple and regularized) and with an RBF-SVM, with BLDA resulting the most accurate classifier. In [89] authors classified two datasets, which are right-hand/right-foot motor imagery and left-hand/right-foot motor imagery, with cross-correlation feature extraction and LS-SVM with RBF kernel. A logistic regression classifier and a kernel logistic regression classifier were used for comparison. LS-SVM achieved higher success rates than the two logistic regression classifiers (first dataset: 95.72% $\pm$ 4.35% versus 89.54% $\pm$ 8.61% and 93.38% $\pm$ 6.76%; second dataset: 97.89% $\pm$ 2.96% versus 95.31% $\pm$ 5.88% and 94.87% $\pm$ 6.98%). In [90] a PCA-based feature selection and RBF-SVM allowed discriminating data from two different datasets, including right-hand and foot motor imagery and left-hand and foot motor imagery, with accuracies ranging from 61.36% to 90.63%. In [91] movement-related independent components and RBF-SVM were used for the classification of left-hand, right-hand and foot motor imagery, with accuracy of 65%. In [92] authors proposed a method based on spatial filtering and 'classifiability' of features for the classification of two different datasets, one consisting of left-hand, right-hand, foot and tongue motor imagery and one of left-hand and right-hand motor imagery, by means of both standard SVM and twin SVM with RBF kernel. For binary classification, accuracy with twin SVM increased by up to 20% with respect to the one obtained by standard SVM. In the multiclass case, accuracy was significantly higher when using twin SVM (up to 79% $\pm$ 5.8% in one subject) against SVM (49% $\pm$ 8.8%). In [93] authors introduced multiclass posterior probability for twin SVM in order to classify both several datasets of motor imagery tasks, made of different classes (from 3 to 11) and a dataset from BCI competitions (left-hand, right-hand, feet, tongue, www.bbci.de/competition/). Twin SVM performed with less computational time and higher accuracy with respect to SVM, especially with fewer samples. In [94] authors applied an optimal allocation-based approach for a discriminative feature extraction to data from binary motor imagery datasets (right-hand versus right-foot and left-hand versus right-foot) and used LS-SVM with RBF kernel and a Naïve Bayes for classification. LS-SVM performed better than Naïve Bayes (accuracy of 90.60% $\pm$ 11.31% and 96.62% $\pm$ 3.72% versus 75.56% $\pm$ 22.35% and 96.36% $\pm$ 2.32%, with 6 and 11 features respectively). In [95] authors optimized common spatial patterns (CSP) computed on multiple signals filtered at a set of overlapping bands, and used RBF-SVM in order to discriminate the imagination of right-hand and foot and of right-hand and left-hand. Results showed a lower error rate when an optimized filter was used (7.95% $\pm$ 2.45% versus 13.33% $\pm$ 2.92% of simple CSP in the first dataset; 18.83% $\pm$ 3.55% versus 23.10% $\pm$ 5.04% of simple SVM in the second dataset). In [96] the imagination of two different motor tasks (slow and fast right-arm flexion) and error potentials were classified by means of two linear SVMs with an average error rate lower by 14% than the case with no error potentials. The imagination of wrist extension was classified in [97], with a linear SVM used for recursive feature elimination and LDA for classification.

Besides classical motor imagery patterns, also parameters related to the imagined movements were discriminated, as in [98], where a wavelet-based feature extraction and an RBF-SVM were used to classify brain signals generated by variations of force-related parameters, during four different voluntary tasks. The error rate provided by RBF-SVM, when compared with a simpler classifier (nearest representative classifier, NR), was significantly lower (15.8% in the best SVM case versus 40.2% in the best NR case). In [99] authors investigated the discriminability of real and imaginary isometric plantar-flexion of the right-foot at different target torques (TT) and at different rates of torque development (RTD), by using wavelets for feature extraction and an RBF-SVM for classification. Results showed that the TTs under ballistic and moderate RTD were discriminated with error rates of 16% $\pm$ 9% and 26% $\pm$ 13% respectively, while RTDs under high and low TT were discriminated with error rates of 16% $\pm$ 11% and 19% $\pm$ 10%, respectively, thus indicating the possibility to detect task parameters in single-trial EEG.

Finger movement imagery was widely used in SMRs-based BCI, as in [100] where authors combined different spatio-temporal brain patterns, different spatial filtering and a linear SVM to discriminate left and right-hand fingers movements, with an accuracy higher than 85%. In [101] an SMRs-BCI based on the discrimination of right-index finger lifting versus left-index finger lifting tasks was proposed. Four different classifiers were compared, namely a Fisher's LDA, two types of ANN and an RBF-SVM which was the best performing classifier on average. In [102] the author used two different datasets, one consisting in the imagination of left- and right-hand movements and one in left- and right-finger lifting movements, and implemented a wavelet-based methodology to localize brain responses in the time-frequency domain, a fractal feature extraction and an RBF-SVM classification. Also he compared RBF-SVM with LDA and demonstrated the former to be more accurate than the latter (82.5% versus 78.7%). Finger movements were also discriminated in [103] by using RBF-SVM, with an average accuracy of 77.11% in detecting movements on all possible couples of fingers. In [104] authors proposed Artificial Bee Colony (ABC) for feature selection in order to discriminate left and right finger lifting. Different features were extracted and both GA and ABC were used for selecting the best feature set. An RBF-SVM was used for classification. Average accuracy was 88.8% with ABC and 83.1% with GA. In [105] authors implemented a method for the recognition of real and imagined finger movements. Numerical and symbolic signal regression procedures were used for feature extraction in order to avoid the loss

of information regarding the time localization of features. Then an ANN and an RBF-SVM were used for classification. Results showed that SVM accuracy increased with the increasing of the number of trials (from 45% to 62%), while ANN performed better at single-trial discrimination.

Finally a new couple of motor imagery tasks, swallowing and tongue protrusion, was analyzed in [106] for post-stroke dysphagia rehabilitation. Features based on dual-tree complex wavelet transform and a linear SVM were used for classification. Results showed that average accuracies of 70.89% and 73.79% could by achieved when swallowing and tongue protrusion were classified against idle state in healthy subjects. Also accuracy of 66.40% for swallowing and 70.24% for tongue protrusion were obtained with data from a stroke patient. Finally authors demonstrated that swallowing could be detected from tongue protrusion-models due to the high correlation between their classification accuracies.

Table 2 summarizes the main SVM features of motor imagery-based BCI works. Differently from ERPs-based papers, the second column of the table lists the tasks performed by the subjects rather than the specific protocol.

As table 2 reports, SVM with RBF kernel and in the standard *C*-SVM implementation is the most adopted configuration, also resulting with the best performances in the classification of motor tasks discrimination, having accuracy up to 90% especially in binary protocols. Grid-search and CV are widely used for hyperparameters setting, while CV is the chosen method for performance evaluation. Even if the information about the used tool is frequently missing, LIBSVM is widely used.

*3.3.2. Motor and non-motor tasks.* Combinations of motor and non-motor tasks have been used in SMRs-based BCI. For example the triad left-hand, right-hand and word generation imagery was discriminated in [107], by means of SVM-recursive feature elimination and linear SVM. Obtained accuracies ranged between 60% and 86.9% against 67.1%–90.2% and 66.7%–86.6% for the left/right-hand, left-hand/word generation and the right-hand/word generation pair, respectively. Those three tasks were classified in [108] too, features being extracted by means of an adaptive CSP and classified by means of an RBF-SVM; this combination resulted more accurate on average (65.12%) than stationary CSP (58.25%) and windowed CSP (59.14%). And also in [109] where linear and RBF-Transductive SVMs (TSVMs are recommended when data distributions differ in the training and testing sets, because they make use of both labeled and unlabeled data to build the learning model) were compared with a linear and an RBF-SVM. With smaller training sets, TSVM outperformed simple SVM by 2%–9% of accuracy, leading to a reduction of calibration time. Moreover non-linear TSVM outperformed linear-TSVM with larger datasets. In [110] feature extraction based on signal wavelet decomposition, tensor discriminant analysis and Fisher scoring (to eliminate redundant features) and RBF-SVM were used to discriminate three different datasets, consisting in (1) the imagination of left- and right-hand movements, (2) the imagination of figure perception and mental arithmetic and (3) a memory task. With motor imagery tasks the method performed with an accuracy comparable to CSP (76.3%), while in cognitive and memory tasks and using a broad frequency band (4–45 Hz), it achieved higher accuracy (up to 92.5% and 75.3%) than CSP (74.9% and 56.9%). In [111] two features extraction methods and two classifiers, Bayesian and RBF-SVM, were compared to discriminate spatial navigation from auditory imagery. The features being equal, there was no significant difference in the accuracy achieved by the Bayesian classifier and the RBF-SVM. In [112] the so called Immune Feature Weighted SVM (IFWSVM) was proposed to classify five different mental tasks (baseline, geometric figure rotation, multiplication problem, letter composing and visual counting). The immune algorithm, which sees the objective function as an Antigen and its optimal solution as an Antibody, was introduced to search for the optimal feature weights and the optimal SVM hyperparameters. When compared with a simple Immune SVM (without feature weight), IFWSVM with an RBF kernel attained higher accuracy in all the tasks (e.g. 97.57% versus 95.75% in the baseline, 91.51% versus 89.69% for the rotation and so on). In [113] the movement of a robot was controlled by means of the motor imagery of four different tasks (move right, move left, move forward and no movement), the stopping on reaching the goal position was controlled by means of the P300 and trajectory was adjusted by detecting error potentials. The four tasks were classified by means of an AdaBoost SVM [114], while P300 responses and error potentials were detected by means of linear SVMs. Average accuracies of 79.20%, 81.50% and 80.10% were obtained for motor imagery, P300 and error potentials detectors respectively. Success rate of 95% was obtained in the real-time control of the robot arm. In [115] authors tested the classification of three different mental tasks (spatial navigation, calculation and reading), when subjects interacted with Mixed Reality scenarios. An RBF-SVM and an LDA were compared, with SVM achieving accuracy of 86.59% and LDA of 88.56%. In [116] authors implemented a MKL-SVM as a linear combination of RBF and polynomial kernels, for the discrimination of five mental tasks (relax, visual counting, letter composing, mathematical multiplication, geometric figure rotation) and of cognitive tasks (identification of specific target stimulations within a stream of non-target stimulations). Accuracy of MKL-SVM in discriminating mental tasks was higher than accuracy achieved by single-kernel SVMs, both when all the five tasks were classified together and when 2, 3, or 4 tasks were considered; the same was found for cognitive tasks. In the single-kernel case, RBF-SVM was more accurate than polynomial-SVM. In [117] authors investigated the continuous evaluation of mental calculation as a valuable signal to control a BCI system. Active states and rest states were discriminated by means of an RBF-SVM. Average AUC values up to $0.89 \pm 0.056$ and $0.67 \pm 0.122$ were achieved in each session and intra session respectively.

Table 3 shows the SVM features relative to motor and non-motor tasks-based BCI systems, with second column reporting performed tasks.

According to table 3, SVM with RFB kernel results to be the most adopted and accurate configuration when dealing with both motor and non-motor tasks. Different implementations are tested,

**Table 2.** SVM features in motor tasks-based BCI papers. L = left, R = right, H = hand. Err = error rate. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [79] | LH, RH | 73.65% | Standard | Linear | 2 | — | NA | 10-fold CV | NA |
| [80] | LH, RH | 75% | Standard | RBF | 2 | — | 20-fold CV | 5-fold CV | SVMLIB |
| [81] | LH, RH | Err = 0–23.81% | Standard | RBF | 2 | — | CV | 10-fold CV | SVM3 toolbox |
| [82] | LH, RH | 91% | Standard | RBF | 2 | — | Genetic algorithm | Train-test | NA |
| [83] | LH, RH | Err = 10.17% | Standard | RBF | 2 | — | Low fraction of SV | Train-test | NA |
| | | Err = 12.14% | Fuzzy | RBF | | | | | |
| [84] | LH, RH | 91% | Standard | RBF | 2 | — | NA | Train-test | NA |
| [85] | LH, RH | 75.73% versus 70.86%, 71.52% and 74.11% | Kernel-fisher-posterior-probability LS-SVM | RBF | 2 | — | Fixed | 5-fold CV | NA |
| [86] | LH, RH | 80.8% | Standard | RBF | 2 | — | 5-fold CV | 5-fold CV | NA |
| [87] | LH, RH, feet, tongue | >80% | Standard | Linear | 4 | NA | Fixed | 5-fold CV | Biosig toolbox |
| [88] | LH, RH, R foot, tongue | Kappa coefficient up to 0.68 | Standard | RBF | 4 | NA | 2 × 5-fold CV | 20-fold CV | LIBSVM |
| | | | | | 4 | | 5-fold CV | Train-test | |
| [89] | RH/R foot | 95.72% ± 4.35% | LS-SVM | RBF | 2 | — | Grid-search | 10-fold CV | LS-SVMLab for Matlab |
| | LH/L foot | 97.89% ± 2.96% | | | | | | | |
| [90] | RH/foot | 61.36% to 90.63% | Standard | RBF | 2 | — | Fixed | Train-test | NA |
| | LH/foot | | | | | | | | |
| [91] | LH, RH, foot | 65 | Standard | RBF | 3 | NA | NA | 5-fold CV | NA |
| [92] | LH, RH, foot, tongue; | 80% | Twin SVM | RBF | 4 | NA | Grid-search | 5-fold CV | LIBSVM |
| | LR, RH | 100% | | | 2 | | | | |
| [93] | Different datasets with 3–11 Classes; LH, RH, foot, tongue | Up to 100% | Standard | RBF | 3–11 | OVA | k-fold CV | Train-test | Matlab |
| | | | Twin SVM | | 4 | | | | |
| [94] | RH/R foot | 96.62% ± 3.72% | LS-SVM | RBF | 2 | — | Grid-search | 10-fold CV | LS-SVMlab toolbox |
| | LH/L foot | 97.39% ± 4.77% | | | | | | | |
| [95] | RH/foot | Err = 7.95% ± 2.45% | Standard | Linear | 2 | — | CV | 10 × 10 CV | NA |
| | RH/LH | 18.83% ± 3.55% | | | | | | | |
| [96] | Arm flexion | 88% | Standard | Linear | 2 | OVA | Online learning | LOO | NA |
| [98] | Force-related parameters | 84.2% | Standard | RBF | 4 | Binary | 3-fold CV | 3-fold CV | NA |
| [99] | Plantar flexion R foot | Err = 16% ± 9% | Standard | RBF | 2 | — | 3-fold CV | 3-fold CV | NA |
| [100] | L/R finger | >85% | Standard | Linear | 2 | — | 5-fold CV | 20-fold CV | LIBSVM |
| [101] | L/R finger | 77.3% | Standard | RBF | 2 | — | Fixed | Train-Test | NA |
| [102] | L/R hand | 82.5% | Standard | RBF | 2 | — | NA | 5-fold CV | NA |
| | L/R finger | | | | | | | | |
| [103] | Fingers | 77.11% | Standard | RBF | 2 | — | Grid-search | 5-fold CV | LIBSVM |
| [104] | L/R finger | 88.8% | Standard | RBF | 2 | — | 5-fold CV | k-fold CV | NA |
| [105] | Thumb/ index | 62% | Standard | RBF | 4 | OVO | Grid-search | Train-test | LIBSVM |
| [106] | Swallowing | 70.89% | Standard | Linear | 2 | — | Fixed | 10-fold CV | Matlab |
| | Tongue protrusion | 73.79% | | | | | | | |

**Table 3.** SVM features in motor-non motor tasks-based BCI papers. R = Right, L = Left, H = Hand, WG = word generation. AUC = area under the ROC curve. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [107] | RH, LH, WG | 60% and 86.9%<br>67.1%–90.2%<br>66.7%–86.6% | Standard | Linear | 3 | OVO | Fixed | Train-test | NA |
| [108] | RH, LH, WG | 65.12% | Standard | RBF | 3 | NA | 20-fold CV | Train-test | NA |
| [109] | RH, LH, WG | >70% | Standard<br>Transductive SVM | Linear<br>RBF | 3 | OVA | 5-fold CV | Train-test | LIBSVM |
| [110] | RH/LH | 76.3% | Standard | RBF | 2 | — | 5-fold CV | Train-test | NA |
| | Figure perception/ mental arithmetic | 92.5% | | | 2 | | | 1Train-4tests CV | |
| | Memory task/no task | 75.3% | | | 2 | | | 1Train-4tests CV | |
| [111] | Spatial navigation/ auditory imagery | 72.2% | Standard | RBF | 2 | — | 5-fold CV | Train-test | Matlab |
| [112] | Baseline, geometric figure rotation, mathematical multiplication, letter composing, visual counting | 97.57%<br>91.51% | Immune Feature weighted | RBF | 5 | OVA | Immune algorithm | Train-test | NA |
| [113] | Robot movements (L, R, forward, no movement) | 79.20%, 81.50%, 80.10% | Adaboost | — | 4 | OVO | NA | Train-test | NA |
| | | 95% online | Standard | Linear | 2 | — | NA | Train-test | |
| [115] | Spatial navigation, mental calculation, mental reading | 86.59% | Standard | RBF | 3 | OVO | Fixed | 7-fold CV | LIBSVM |
| [116] | Relax, visual counting, letter composing, mathematical multiplication, geometric figure rotation | 99.20% | MKL | Polynomial | 2 | — | Multiple values search | 5-fold CV | Matlab |
| | | 81.25% | | RBF | 3 | OVA | | | |
| | | 76.76% | | | 4 | OVA | | | |
| | | 75.25% | | | 5 | OVA | | | |
| [117] | Continuous mental calculation | AUC up to $0.89 \pm 0.056$ | Standard | RBF | 2 | — | 20-fold CV | CV | LIBSVM |

even if the standard one is still the most used. CV is often adopted for hyperparameters setting, while train-test is usually considered for performance evaluation. Again, LIBSVM is widely used, even if the information about the tool is frequently omitted.

*3.3.3. Other tasks.* Imagery tasks different from the classical ones were used in [118], where the imagery of the 'yes' and 'no' words was discriminated by means of an RBF-SVM with accuracy higher than 70%. In [119] authors designed a protocol based on the imagination of two Chinese characters and used RBF-SVM to discriminate them with high accuracy (from 73.65% to 95.76%). In [120] a three-layer scheme for emotion recognition in single-trial EEG was proposed. Emotion-inductive pictures were used and valence and arousal were classified by means of imbalanced (quasi-conformal) SVM with RBF kernel. Results showed that the proposed scheme could achieve the highest classification accuracy of valence (82.68%) and arousal (84.79%) when compared to k-Nearest Neighbors (kNN) or standard SVM. In [121] a real-time algorithm for the classification of EEG-based self-induced emotions (disgust and relax) was proposed. RBF-SVM was used for classification, achieving average accuracy higher than 90%. In [122] authors proposed a method for the recognition of implicit human intentions for developing an interactive web service engine. Brain state changes associated to navigational and informational intention were measured by using EEG phase synchrony values in different frequency bands as features. Different classifiers were compared: SVM with RBF and polynomial kernels, Gaussian mixture models (GMM) and Naïve Bayes. Classification accuracy was the highest for RBF-SVM: for example, for one subject, accuracy with RBF-SVM was 77.4% whereas with polynomial-SVM, Naive Bayesian and GMM it was 72.4%, 52.2% and 50.5%. The same trend was observed in all subjects.

In table 4 SVM features relative to the 'other tasks'-based BCIs are listed.

**Table 4.** SVM features in other tasks-based BCI papers. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [118] | YES/NO imagination | >70% | Standard | RBF | 2 | — | Fixed | 10-fold CV | OSU SVM Classifier |
| [119] | Chinese characters imagination | 73.65%–95.76% | Standard | RBF | 2 | — | 10-fold CV | 10-fold CV | LIBSVM |
| [120] | Emotion reconition | 82.68%–84.79% | Imbalanced quasi-conformal kernel SVM | RBF | 2 | — | 2-fold CV | CV | NA |
| [121] | Emotion reconition | Up to 99.4% ± 0.6% | Standard | RBF | 2 | — | 10-fold CV | Train-Test | LIBSVM |
| [122] | Human intention recognition | 77.4% | Standard | RBF Polynomial | 2 | — | Grid search + 10 fold CV | 10-fold CV | NA |

**Table 5.** SVM features in SSVEPs/SCPs-based BCI papers. NA = not available.

| Paper | Brain Feature | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [123] | SCPs | Up to 97.39% | Semisupervised 1-norm | Linear | 2 | — | NA | Train-test | LP-solve 2.0 C library |
| [124] | SCPs | Up to 88.6% ± 1.9% | Standard | RBF | 2 | — | Fixed values for $C$ CV for $\sigma$ | CV | Matlab |
| [125] | SCPs | 89.12% | Standard | Sigmoid | 2 | — | NA | 10 × 10-fold CV | NA |
| [68] | SSVEPs | Up to 100% | Standard | Order4-polynomial, quadratic, linear | 4 | OVA | NA | Train-test | Matlab |

RBF-SVM in its standard configuration is widely used for the classification of unconventional imagery tasks, with high accuracy. CV is widely used both for hyperparameters setting and for performance evaluation.

### *3.4. Steady states visual evoked potentials (SSVEPs) and slow cortical potentials (SCPs)-based BCI*

For the classification of SCPs, Qin *et al* [123] proposed a semi-supervised SVM aiming at reducing the time-consuming training process. The method consisted in using both a small labeled dataset and a large unlabeled dataset to train the classifier and in implementing a batch-mode incremental training to iteratively improve the training performances. A 1-norm linear SVM further decreased training time. The method was validated with data from an EEG-based cursor control experiment and CSP was adopted for data filtering. The proposed semi-supervised SVM increased the accuracy by more than 14% with respect to a standard SVM. Moreover, the accuracy reduced by 3.25% when the standard SVM was trained with all the data (labeled plus unlabeled), but a decrease in CPU time was achieved. In [124] a method to improve BCI accuracy was proposed, based on polynomial fitting of training data and the use of a very simple kNN. This was made to discriminate brain potentials generated when moving a cursor up and down on the screen. An RBF-SVM and an ANN were also compared to the simple kNN. The latter outperformed both SVM and ANN in terms of accuracy and speed. In [125] an automated feature selection strategy, based on a decision tree, was implemented. Subjects were asked to move a cursor up and down on a screen and their SCPs were classified by means of a sigmoidal-kernel SVM, which performed with an average accuracy of 89.12%. This result was by 0.9% higher than the one obtained with an optimal electrode recombination method for feature extraction.

For SSVEPs classification, instead, in [68] authors developed a prototype of BCI-controlled wheelchair. SSVEPs elicited by four different flickering frequencies were used to control the movement of the wheelchair in four directions. Four colors (green, red, blue and violet) were used for the stimuli to investigate the influence of colors on SSVEPs. Two ANNs and an SVM were used as classifiers; different kernels were tested (4th order-polynomial, quadratic, linear) in order to select the most accurate one. On the basis of the results, when using violet stimuli, SVM achieved the best accuracy (between 75% and 100%).

Table 5 reports SVM features relative to SCPs and SSVEPs-based BCI systems. In particular, the second column lists the exploited brain feature (SCPs or SSVEPs).

From table 5 it is evident that different kernels are used to set a standard SVM in order to classify SSVEPs and SCPs, with almost perfect accuracy. The information about the hyperparameters setting is often missing.

### *3.5. Some statistics*

Figure 7 represents the percent occurrence of SVM features related to papers focused on BCI and here reviewed. The standard $C$-SVM implementation and the RBF kernel result as the most popular. Moreover, when dealing

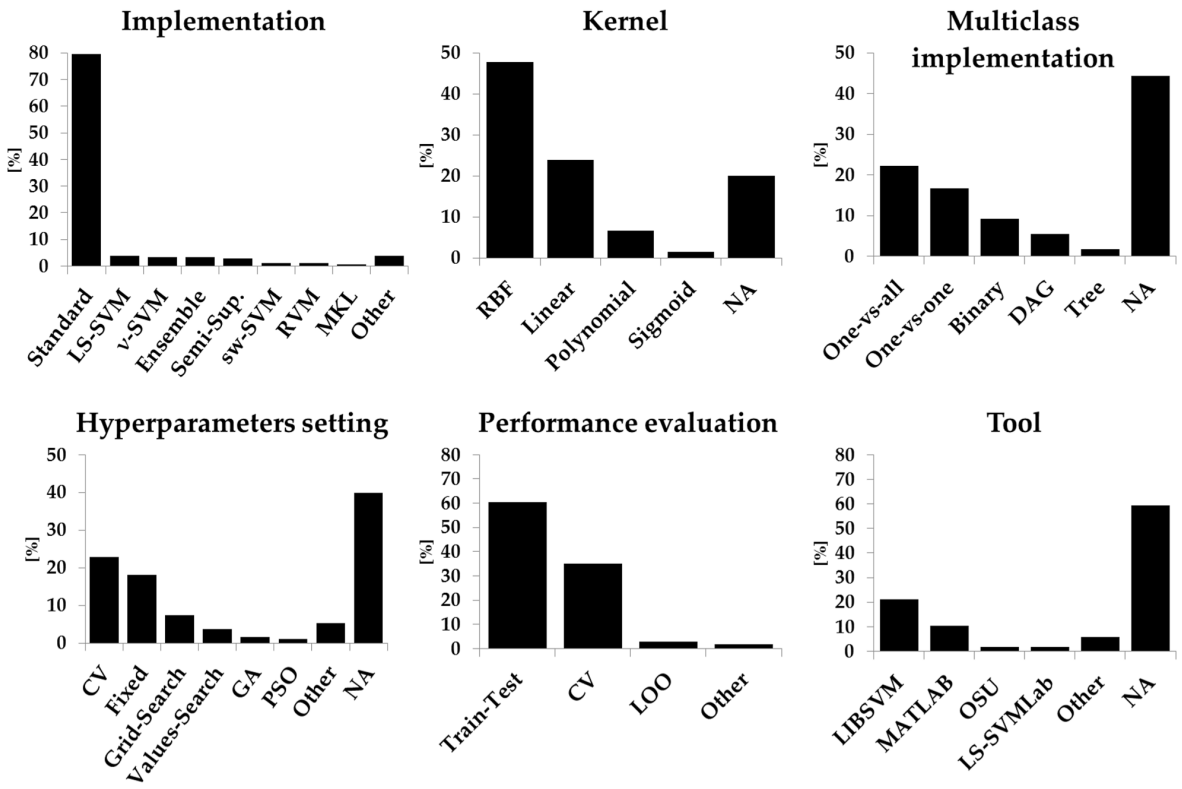


**Figure 7.** Percent occurrence of the different SVM features (Implementation, Kernel, Multiclass Implementation, Hyperparameters setting, Performance evaluation, Tool). Values are obtained from literature works regarding EEG-based HCIs. LS = least-squares; sw = spatially-weighted; RVM = relevance vector machine; MKL = multiple kernel learning; RBF = radial basis function; DAG = directed acyclic graph; CV = cross validation; GA = genetic algorithm; PSO = particle swarm optimization; LOO = leave-one-out; NA = not available.

with multiclass problems, the OVA approach is preferred. Concerning hyperparameters setting, the CV is the most frequent method, whereas the performance evaluation adopts a train-test approach in most of the cases. Among tools, LIBSVM results the mostly utilized one. However, it is important to stress that, despite the easy accessibility of all the information regarding SVM algorithms, 20% of the available papers (not reported here) did not described the kernel type, even if its choice dramatically affects the final results. Moreover, in most of the reviewed papers, the multiclass implementation and the hyperparameters setting methodology are not described, thus leading to only partially reproducible results.

## 4. EMG-based HCI and SVM

### *4.1. Overview of EMG-based HCI systems*

Surface EMG (sEMG) signals are the result of the capture of the electrical activity of muscles, recorded on skin surface. Similarly to BCI systems, sEMG-based HCIs found their major applications in the field of assistive devices and rehabilitation. They have been mainly exploited to control multifunction prostheses of the upper limb [126] or the lower one [127]; to drive electric power wheelchairs [128]; to recognize facial gestures [129]; to control a mobile robot [130] or a manipulator system [131]; to implement functional electrical stimulation-control systems for hemiplegic patients rehabilitation [7], and so on.

In the following sections a review of all the literature studies based on SVMs to implement HCI systems is provided. The focus is mainly on HCI used for controlling purposes, thus disregarding, for example, papers using SVM to detect neuromuscular disorders [132, 133], to discriminate contaminated from clean EMG [134], to discriminate efforts stages in prolonged running [135].

Papers are divided according to the body segment involved and/or the tasks to be recognized: hand movements, finger movements, arm movements, walking modes, facial and whole body movements.

### *4.2. EMG-based HCI for hand gestures recognition*

Hand gestures recognition is one of the most useful applications of sEMG-based HCI systems for the control of robotic hands as reported in [136], where wavelet transform for feature extraction and a linear SVM allowed to discriminate six hand movements with a misclassification rate of 5%. In [137] six hand motions and the relative forces were recorded by means of EMG and force sensors and classified with RBF-SVM. Classification accuracy was up to 95%, while error rate was lower than 7% in force regression. In [32] authors developed an EMG-controlled humanoid hand to discriminate active modes from idle modes

and then eighteen different hand gestures within the active modes. An RBF-SVM was used and accuracy close to 100% was obtained. For the online design, the RBF kernel was substituted with a linear one to relieve the computational burden, with a slight decrease in accuracy. In [138] fifteen different hand gestures were recorded in senior and young people and classified with an RBF-SVM with accuracy of 90.62% (seniors) and of 97.60% (young volunteers) respectively. In [139] ten hand-performed Chinese numbers were classified. Three feature sets and four classifiers (kNN, LDA, quadratic discriminant analysis (QDA), and SVM) were combined and compared; moreover the combination of the three features and of an MKL-SVM, with RBF and polynomial kernels, was investigated. The latter was the most accurate combination, with average accuracy of 97.93%. In [25] authors compared supervised and unsupervised data processing for the classification of five hand gestures in both healthy and amputee subjects. Two preprocessing algorithms were compared, PCA and CSP, together with different extracted features and three different classifiers, namely ANN, RBF-SVM and LDA. Results in both healthy and amputee subjects showed no significant difference in accuracy when using CSP or PCA, and that ANN was more accurate than SVM and LDA. In [140] six different hand motions were recorded and different combinations of feature extraction and classifiers (simple logistic regression (SLR), decision tree (DT), logistic model tree (LMT), ANN, LDA, and SVM) were compared. Results showed an error rate lower than 15% when using time-domain features. SLR and LMT outperformed all the other classifiers. SVM achieved error rates similar to those of the other classifiers (10% with time domain features). In [141] data relative to seven different hand motions were recorded and a combination of autoregressive model coefficients and time-domain features was used as feature set. Incremental-learning adaptive SVM was implemented for classification, consisting in online incorporating useful information from testing data into the classification model, thus creating an incremental learning. When compared to a traditional SVM, the adaptive SVM obtained performances higher by 3.3% and 8% in intra-session and inter-session tests respectively.

Table 6 reports SVM features relative to EMG-based hand movements recognition papers.

From table 6 it can be seen that even with a high number of classes, SVM achieves accuracy near 100%. RBF-SVM in standard implementation is the most used configuration; moreover, there is not a univocal choice about multi-class implementation, while CV is the preferred method for both hyperparameters setting and performance evaluation. LIBSVM occurs many times as the used toolbox.

### 4.3. EMG-based HCI for finger movement recognition

SVM performed with high accuracy in the recognition of finer finger movements, as reported in [142], where forearm sEMG, force/torque sensors and an optimized machine learning technique allowed on-line control of both finger positions and finger force of a dexterous robotic hand. Three classifiers were compared, ANN, RBF-SVM and locally-weighted Projection Regression, in the discrimination of five grasping modes. For force detection, the regression produced one single output, that is, the target force value. With healthy subjects, grasping modes were classified with an accuracy of 89.67% $\pm$ 1.53%, while the applied force was predicted with an average error of 7.89% $\pm$ 0.09%. The RBF-SVM resulted only marginally better than the other classifiers. Finger motions were also discriminated in [143], where a method for optimizing the online learning of SVMs was introduced. Such method consisted in updating the constant term of the decision function whenever a misclassification occurred. The optimized SVM, with RBF kernel, was compared with a classical SVM and an ANN, in discriminating six different finger motions. Results showed an increase in accuracy when using the proposed SVM (83.6%–87.1%) against ANN (75.5%–77.3%) and simple SVM (75.9%–82.8%). In [144] authors proposed twin SVMs for the classification of seven tasks involving finger and wrist flexion and compared them with ANN. Results showed that the performances of ANN degraded with the increasing of the number of classes (99.34% in accuracy with three classes versus 59.34% in accuracy with seven classes). When using twin SVMs with linear and polynomial kernels, accuracy reached up to 85.07%; with a sum of RBF kernels, accuracy was 86.94%, while with the sum of three different kernels (linear, polynomial and RBF) accuracy was as high as 87.27%. In [7] a muscle-computer interface was designed to detect drivers' movements and avoid their distraction. Twelve classes of finger pressure and 2 of finger pointing were recorded and seventeen different features were extracted from each channel and reduced by means of a fuzzy neighborhood discriminant analysis. Several classifiers were compared, namely RBF-SVM, LDA, Regression Tree and Naïve Bayes classifier. SVM achieved the lowest average error rate (7%).

Table 7 reports SVM features relative to EMG-based finger movement recognition papers.

For finger movement recognition, RBF-SVM is used with high accuracy, even in case of a huge number of classes (up to 14).

### 4.4. EMG-based HCI for arm movement recognition

Many EMG-based systems are related to the control of the movements of a robotic arm. In [145] authors proposed cascading generalized discriminant analysis for dimensionality reduction, optimal features selection, and an RBF-SVM for the recognition of eight arm postures for prosthesis control. When compared to kNN or ANN, performance of the cascade-approach was significantly higher both in terms of accuracy (accuracy was 93.54% on average) and of recognition time (20.7 ms). Oskoei *et al* [146] discriminated six classes of limb motions by means of different classifiers, namely SVM with linear, RBF, sigmoid and polynomial kernels, LDA and multi-layer perceptron (MLP)-ANN with 1 or 2 hidden layers. The four applied kernels performed similarly over the considered features, with an average accuracy of 95.5% $\pm$ 3.8%. Also LDA performed similarly to SVM; MLP-ANN with two hidden layers performed similarly to LDA and SVM and finally MLP-ANN with one hidden layer achieved accuracy inferior by 6% with respect to the other classifiers. In [147] eight arm positions were discriminated by

**Table 6.** SVM features in EMG-based HCI for hand gesture recognition. Err = error rate. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [136] | Wrist flexion and extension, hand supination, pronation, opening and closing | 95% | Standard | Linear | 6 | OVA | NA | 3-fold CV | NA |
| [137] | Rest, index pointing, power grasp, precision pinch grip, precision tripodal grip and hand stretching | 95% | Standard $\varepsilon$-SVR | RBF | 6 | OVO | Logarithmic Grid-search + 5-fold CV | 10-fold CV | LIBSVM |
| [32] | Hand gestures | 100% | Standard | RBF linear | 18 | OVO | 4-fold CV fixed | Train-test | LIBSVM |
| [138] | Hand gestures | 90.62% (seniors)<br>97.60% (young) | Standard | RBF | 15 | OVO | Grid-search + 8-fold CV | Train-test | LIBSVM |
| [139] | Hand-performed Chinese numbers | 97.60% | Standard MKL | RBF polynomial | 10 | OVA | Grid-search | LOO | NA |
| [25] | Hand gestures | Up to 97% | Standard | RBF | 5 | OVA | Fixed | 5-fold CV | Matlab |
| [140] | Hand close, hand open, wrist flexion, wrist extension, ulnar deviation (healthy subjects) or hand supination (transradial patients) and radial deviation | Err = 10% | Standard | RBF | 6 | OVA | Fixed | Train-test | WEKA |
| [141] | Rest, hand open, hand close, supination, pronation, wrist flexion and wrist extension | 93.3% ± 4.1%<br>96.6% ± 1.5% | Standard<br>Incremental learning adaptive SVM | Linear | 7 | OVO | Adaptive | 5-fold CV | NA |

**Table 7.** SVM features in EMG-based HCI for finger movement recognition. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [142] | Grasp by opposing thumb and index, thumb and middle, thumb and ring, thumb and all the other fingers, and no grasping | 89.67% ± 1.53% | Standard<br>Regression | RBF | 5<br>1 | NA | Grid-search | 5-fold CV | LIBSVM |
| [143] | Finger motions | 83.6 to 87.1% | Standard + additional learning | RBF | 6 | OVO | NA | Train-test | NA |
| [144] | Fingers and wrist flexion | 87.27% | Twin SVM | Linear, polynomial, RBF | 7 | OVA | Optimization scheme | 10-fold CV | Matlab |
| [7] | Finger pressures and finger pointing | 93.9% ± 4.4% | Standard | RBF | 14 | NA | Fixed | Train-test | LIBSVM |

means of a linear SVM, with accuracy within 92% and 98%, while in [148] a four-level wavelet transform was investigated as a novel kernel for an LS-SVM aiming at classifying four different limb motions. Accuracy >90% was obtained with just ten features, better than the one found with a MLP, especially with small training sets. Khokhar *et al* [149] explored the possibility to control the torque applied by the wrist by means of sEMG classification and of a wrist exoskeleton prototype in real-time. An RBF-SVM was employed to discriminate nineteen different force intensities, with average accuracy of 88.20% ± 3.45%. When reducing the motion set to thirteen classes, accuracy raised to 96.52% ± 1.98%. In [150] pointing movements (north, south, west and east) on a horizontal plane while holding a robotic manipulandum

**Table 8.** SVM features in EMG-based HCI for arm movement recognition. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [145] | Arm postures | 93.54% | Standard | RBF | 8 | OVO | Grid search | Train-test | NA |
| [146] | Limb motions | 95.5% ± 3.8% | Standard | Linear RBF sigmoid polynomial | 6 | OVO | 5-fold CV | 5-fold CV | LIBSVM |
| [147] | Arm positions | 92% and 98% | Standard | Linear | 8 | OVO | Across-session CV | 5-fold CV | LIBSVM |
| [148] | Fist clench, fist stretch, wrist pronation and wrist supination | Up to 100 % | LS-SVM | Wavelet packet transform | 4 | OVA | NA | Random train-test | NA |
| [149] | Wrist motions | 88.20% ± 3.45% | Standard | RBF | 19<br>13 | OVO | Grid-search + 8-fold CV | Train-test | LIBSVM |
| [150] | Movement directions (north, south, west and east) | 90.62% | Standard | RBF | 4 | NA | Fixed | Iterative train-test | LIBSVM |
| [151] | Arm movements | Up to 90% | $\nu$-SVR | RBF | 14 | NA | Fixed | 4-fold CV | LIBSVM |
| [152] | Wrist motions | 80.57%<br>85.69% | Standard | RBF<br>Polynomial | 5 | NA | NA | 10-fold CV | NA |
| [153] | Shoulder flexion and extension, elbow flexion and extension | 93.9% ± 4.3% | Standard | RBF | 4 | OVO | 5-fold CV | Train-test | LIBSVM |
| [154] | Hand/wrist motions | 93.22% ± 4%<br>93.28% ± 7% | Fuzzy RVM<br>Fuzzy LS-SVM | — | 6 | OVO | NA | Train-test | Matlab |

were discriminated with an RBF-SVM. With healthy subjects, an average accuracy of 93.9% ± 4.4% was achieved; with stroke subjects, accuracy dropped to the 30%–70% range, when a subset of muscles was used, and to the 36.7%–83.3% interval when all muscles were used. In [151] authors proposed a system for real-time myoelectric control of multiple degrees of freedom (DOFs), including wrist flexion-extension, abduction-adduction and forearm pronation-supination. Both healthy subjects and amputees took part to the experiment, performing fourteen different arm movements aimed at moving a cursor on a screen. Two classifiers were compared, an ANN and an RBF-SVM-based algorithm using $\nu$-SVR. For the healthy subjects, SVM significantly outperformed ANN in throughput, completion rate, overshoot and path efficiency. For the amputees, SVM outperformed ANN in path efficiency and throughput with the first amputee and in throughput with the second one. SVM significantly reduced the computational time for both training and real-time control. In [152] authors proposed a Deep Belief Network for the classification of five different wrist motions. When compared to LDA, SVM with RBF and polynomial kernels and back-propagation classifier, the proposed classifier achieved significantly higher accuracy (89.95% versus 83.74%, 80.64%, 87.66% and 89.53%, respectively) and lower training time. In [153] a new feature ranking method, based on short-time Fourier transform, was proposed for the classification of two series of shoulder and elbow motion patterns driven by an exoskeleton robot arm. The proposed feature extraction method was compared to classical time-domain and frequency-domain approaches. RBF-SVM was used for classification. Results showed that with the ranking feature approach, the accuracy was greater than with conventional features (93.9% versus 33.3%–90.8%). In [154] authors compared the performances of fuzzy-RVM and LS-FSVM for the discrimination of six classes of hand/wrist motions. Both time-domain and frequency domain features were extracted and compared. Results showed that FRVM achieved accuracy similar to FSVM with time domain features (93.22% ± 4% versus 93.28% ± 7%) and frequency-domain features (90.86% ± 6% versus 92.81% ± 4%). On the other hand, processing delay was significantly lower for FRVM than FSVM (e.g. 34.47 ms ± 29.36 ms versus 123.46 ms ± 54.35 ms with time domain features). Training time was significantly higher in FRVM than in FSVM.

Table 8 summarizes SVM features relative to EMG-based arm movement recognition papers.

From table 8 it can be seen that SVM with RBF kernel and standard implementation is the most used configuration and achieves high accuracy even in case of a high number of classes. OVO is preferred for multiclass implementation, CV for hyperparameters setting and train-test for performance evaluation. Again, LIBSVM is the most used toolbox.

### *4.5. EMG-based HCI for walking modes recognition*

In [155] authors developed an algorithm aimed at recognizing different locomotion modes performed by transfemoral amputees, by means of sEMG and force/moments measurements. Six locomotion modes and five transition modes were recorded, while maximum/minimum/average amplitudes of force or moment were measured as the mechanical signal features. LDA and RBF-SVM were compared. Joining EMG and mechanical features in the discrimination of static states resulted with accuracy higher than when using one-mode features. SVM's

**Table 9.** SVM features in EMG-based HCI for walking mode recognition. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [155] | Locomotion modes; transition modes | 96.52% ± 1.98% | Standard | RBF | 5<br>5 | OVO | Fixed | LOO | NA |
| [156] | Stair ascent and descent, ramp ascent and descent, level ground walking at three different speeds | 97.90% ± 1.39% | Standard | Linear | 7 | OVO | NA | k-fold CV | Matlab |

**Table 10.** SVM features in EMG-based HCI for facial and body movement recognition. NA = not available.

| Paper | Tasks | Performances | Implementation | Kernel | Nr. of classes | Multiclass implementation | Hyperparameters setting | Performance evaluation | Tool |
|---|---|---|---|---|---|---|---|---|---|
| [157] | Pointing task towards a target under unconstrained, knee extended, reduced base of support, imposed straight finger trajectory and imposed semicircular finger trajectory | 80% | Standard | Wigner | 5 | NA | NA | 5-fold CV | Matlab |
| [158] | Wheelchair control states | 90–93%<br>73–81% | Adaptive Incremental online;<br>Standard | RBF | 2 | — | Fixed | Train-test | LIBSVM |
| [159] | Facial gestures | 93.1% ± 1.30% | LS-SVM<br>Standard | RBF<br>Linear | 10 | OVO<br>OVA | Manual<br>Automatic | LOO<br>k-fold CV<br>generalized CV | LS-SVMlab toolbox |

accuracy was higher than LDA's one by 1.5%–5.9%. Also in the identification of transition modes, fusion-based SVM's accuracy was higher than EMG only-based SVM and fusion-based LDA's accuracies. In [156] authors implemented a classification algorithm to discriminate myoelectric walking modes in transtibial amputees from sEMG, in order to improve control of prostheses. An LDA and a linear SVM were compared and their average accuracies resulted similar both in the amputees (97.90% ± 0.22% versus 97.90% ± 1.39%) and in the non-amputees (93.30% ± 2.62% versus 94.70% ± 2.82%).

Table 9 reports SVM features relative to EMG-based walking modes recognition papers.

### *4.6. EMG-based HCI for facial and body movement recognition*

In [157] authors used an SVM with Wigner kernel (the Wigner kernel is defined as $K(x_i, y_i) = \left|\left\langle x_i, y_i \right\rangle\right|^2$, with $<.,.>$ denoting the inner product) for the discrimination of five body pointing tasks with or without postural or focal constraints. A mean accuracy close to or higher than 80% was achieved in discriminating constrained from unconstrained movements. Xu *et al* [158] presented an sEMG-based HCI for the hand-free control of an intelligent wheelchair. Forehead sEMG signal was recorded and two control signals were produced, namely single jaw click and double jaw click, which allowed selecting five different control states designed for the wheelchair (forward, backward, left, right and stop). An online incremental SVM, with RBF kernel, was used for classification, and achieved higher accuracy and lower training time than a simple SVM. In [159] an EMG-based HCI for the recognition of ten different facial gestures was implemented. Different models of multi-class LS-SVM were designed and kernel parameters were tuned both manually and automatically. In particular authors compared linear and RBF kernels, different model of validation techniques (e.g. LOO-CV, k-fold CV and Generalized CV) and different encoding schemes (e.g. OVO, OVA, etc.). Results showed that when parameters were automatically tuned, the most accurate model included RBF as kernel and OVO as encoding scheme (accuracy 88.90% ± 2.16%). With manual tuning, a model containing RBF as kernel and OVA as encoding scheme was the most performing (accuracy 93.1% ± 1.30%). As a result, RBF was the most accurate kernel at the price of high computational time. LS-SVM resulted also the most performing classifier (93.1% ± 1.30%) when compared to standard SVM (85.50% ± 2.85%), fuzzy c-means (90.41% ± 3.12%) and fuzzy Gath–Geva clustering (91.82% ± 2.71%).

Table 10 shows SVM parameters for EMG-based body and facial movement recognition papers.

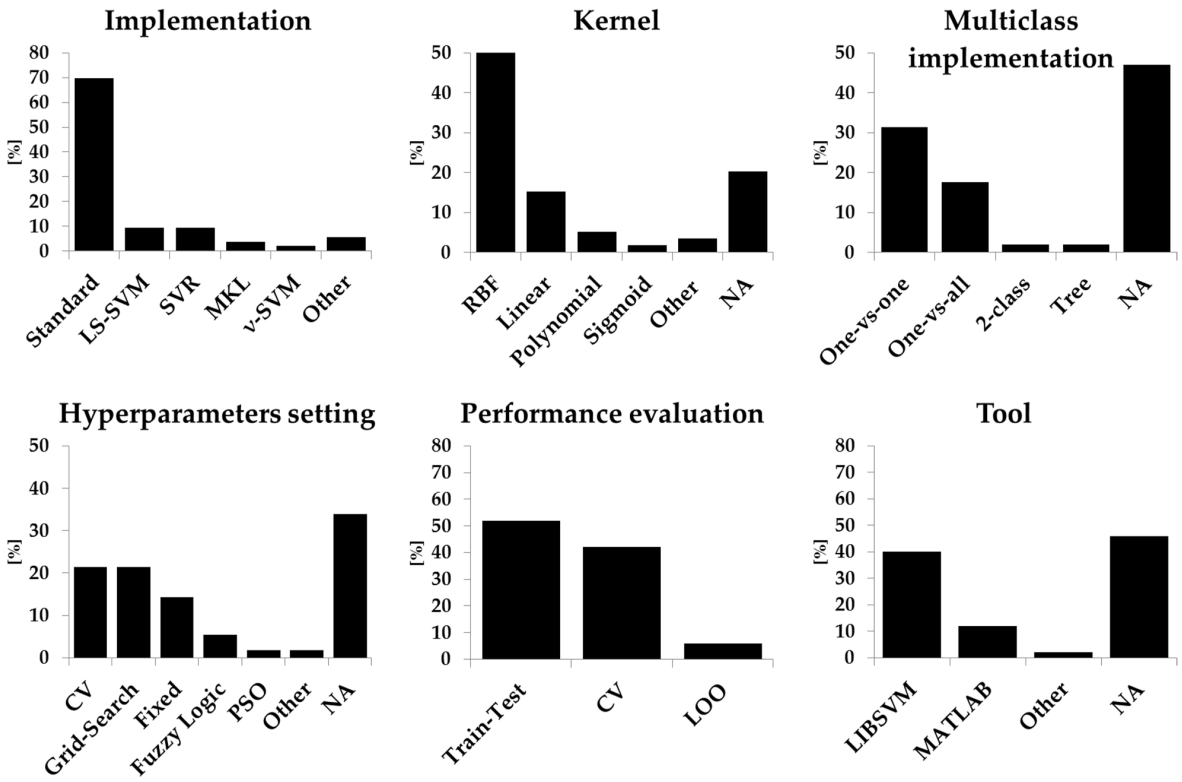


**Figure 8.** Percent occurrence of the different SVM features (Implementation, Kernel, Multiclass Implementation, Hyperparameters setting, Performance evaluation, Tool) in the literature of EMG-based HCIs. LS = least-squares; SVR = support vector regression; MKL = multiple kernel learning; RBF = radial basis function; CV = cross validation; GA = genetic algorithm; PSO = particle swarm optimization; LOO = leave-one-out; NA = not available.

### *4.7. Some statistics*

Figure 8 shows the percent occurrence of the SVM characteristics reviewed across EMG-based HCI papers. In accordance to the EEG-based HCI works, the standard *C*-SVM with RBF kernel results to be the most adopted implementation. Then the multiclass OVO approach is preferred (please note that, in general, such systems are based on a larger number of tasks to be recognized with respect to the EEG-based systems). CV and grid-search and train-test modalities are mostly adopted for hyperparameters setting and performance evaluation, as in the EEG case. LIBSVM is the most used tool.

## 5. Discussion

Given the great advantages of SVM for EEG-based and EMG-based HCI systems, a comprehensive review of related works can be useful to develop optimized analysis strategies and correctly design up-to-date systems. The aim of this paper is exactly to cover all the main aspects at the base of SVM-classification for HCI.

Concerning EEG-based HCI, SMRs are considered to be more complex brain signal features than event-related/evoked responses (P300 and SSVEPs), as they are not event-locked and need complex preprocessing before classification (SSVEPs are dominant in the EEG spectrum and ERPs can be detected after a simple signal averaging). Moreover, SMRs amplitude increases after intensive subject training, whereas SSVEPs and ERPs features can be identified since the first use of the system. As a consequence, SMRs detection needs algorithms able at working with complex datasets and not-enhanced features, while SSVEPs are sometimes detectable with a threshold method and ERPs are sufficiently discriminable with a simple LDA. Due to their robustness, SVMs are highly preferred for SMRs-based BCI data classification.

In general, our review underlines that, when different classification approaches are compared, SVM outperforms the others in terms of accuracy [68, 75–77, 80–83, 87, 89, 98, 101, 102, 105, 122]. However a minority of works report SVM with lower performance than other classifiers [88], even if the statistical significance of such difference is not assessed [115].

With respect to kernels, RBF usually outperforms linear or polynomial ones [76–78]. All considered, SVMs adopting RBF kernel result to be suitable classifiers for detecting brain responses in EEG-based BCIs. This is more evident with SMRs-based paradigms and with large datasets.

When dealing with EMG-based HCIs, special care is given to the online performances of the classifiers, due to the real-time requirements of prosthesis/device control. When different classification approaches are considered, SVM outperforms other classifiers (generally LDA, ANN or kNN)

in most of the cases [11, 76, 139, 142–144, 148, 151, 159]. Differently in [25, 146] SVM achieves accuracy worse than ANN and in [152] SVM performs worse than Deep Belief Networks. In case of different kernels, usually RBF results the most accurate one, whereas in [146] linear, RBF, polynomial and sigmoidal kernels achieve similar performances.

In general SVM, mainly with RBF kernel, performs with excellent accuracy even in case of a high number of tasks to be recognized, as in [11, 32, 138, 149, 151]. Also the low recognition time, typical of some particular SVM implementations, as pointed out in [145, 149, 151, 158], makes SVM preferable for real-time control.

As a result, with respect to other classifiers commonly employed in the HCI literature for the classification of physiological patterns, such as LDA and ANN, SVMs have peculiarities that make them recommended when dealing with particular classification scenarios:

(1) *Insensitivity to big amount of features*: SVM is insensitive to the number of features extracted to describe data, thus leading to a reduction of the problem complexity. This does not occur in LDA-based classifiers, which suffer from the 'curse of dimensionality' (see section 2.1): if the number of training samples is less or equal to the number of features, the problem becomes ill-posed (the covariance matrix becomes singular and cannot be inverted) and some regularization strategies are needed [160]. SVM being insensitive to big data size is particularly appreciated in HCI field, especially in the EEG-based case. Indeed, EEG is usually recorded with many electrodes (up to 256), with a high number of features extracted to describe the classification problem. Also, performing BCI training sessions can be particularly stressful and exhausting for the subjects, so it is not uncommon to have only a few training samples available due to subjects' fatigue. Differently, in the EMG-case the problem of dimensionality is less important, as EMG patterns are easily discriminable with few channels (and few features) and EMG recording sessions are easier to afford. Nevertheless SVMs are still recommended for their high performance.

(2) *Generalization*: SVM penalty term $C$ allows controlling the complexity of the model and hence to avoid overfitting, which occurs when a statistical model describes random error rather than the real data structure, decreasing the ability of the classifier to discriminate unseen data (*generalization*). LDA, instead, unless it is provided with some regularization approach, suffers from overfitting if the number of training samples is not sufficient with respect to the feature set size. ANNs usually overfit if training is too long or if the model is too complex (e.g. too many hidden neurons). The overfitting issue is particularly relevant when training data are noisy. This is frequent with brain signals, which are not stationary, and can be contaminated by different sources: eyes' movements, head muscles' contractions, sweating, heart beating, and so on. A robust and outliers-insensitive classifier is therefore needed to deal with such complex signals.

(3) *Unique solution*: training of SVM always finds a global minimum; therefore, the solution to the classification problem is global and unique, whereas ANNs usually converge only to locally optimal solutions. This means that each time a neural network is trained, it can result in a different solution due to initial network weights.

(4) *Training complexity*: usually a time-consuming training phase is necessary in HCI systems in order to allow the classifier to learn the specific characteristics of the signals and reducing the training process is always a challenge. LDA training may result very slow and inefficient in case of big dimensionality of data, due to the presence of matrix inversions and of matrix decompositions (unless regularization strategies are adopted). ANN training may be slow with big data size as well, because the complexity of the network increases. Conversely, if SVMs, especially the kernelized ones, could be slow in training and much slower in testing, a lot of different implementations, which dramatically decrease training burden, exist (see for example LS-SVM or transductive SVMs [109]).

(5) *Suitability for online implementations*: strictly connected to the training complexity is also the suitability of the classifier for online applications. In BCI, for example, online feedback is very important to boost the performances of the subject and the classifier needs to be efficient to not slow down the whole system. In the same way in EMG-based HCI for device control, there is a time limit between the user's command and the prosthesis response that has to be respected to avoid the decay of prosthesis performance (near 100 ms) [161].

In a classical online learning scenario, training data are available in a sequential order, whereas in an offline (batch) mode, all training examples are available at once. In case of very large and non-stationary data, such as physiological data, offline classification algorithms might not be suitable; in fact, with large datasets, training can be computationally expensive, while if different data distributions varying over time are presented and integrated into the learning rule all together, problems can arise for the learning [162]. Hence the necessity of having online implementations for the used classifier, such as the incremental algorithms, which update the solution of a classification problem after one training sample is added to or removed from the training set. A lot of incremental algorithms have been implemented for SVMs in the machine learning literature; see for example [163], where LASVM, an approximate C++ SVM solver that uses online approximation, is presented; or www.cpdiehl.org/code.html, where a Matlab implementation of incremental SVM learning is provided. Anyway, despite their undoubted utility for online applications, a limited interest was shown by the HCI community to incremental SVM, maybe due to the absence of well-accepted implementations, as it happens with LIBSVM or SVMLight for SVM batch learning [162]. In fact, within this review incremental SVMs were used in [141, 158] for EMG-based HCIs and in [123] for BCIs. Incremental algorithms were also proposed for LDA [164] and ANN [165].

Nevertheless the superiority of SVM in HCI online scenarios, with respect to other classifiers such as LDA or ANN, is still a subject matter of discussion. In [166] authors compared accuracies and Matlab runtimes of four different classifiers, LDA, non-linear SVM, BDLA and batch-perceptron, in the classification of P300-speller data, in order to test their feasibility for online implementation on a small digital signal processing board; LDA showed the slowest runtime (60 s versus 14.5 s of BLDA) and also an average runtime twice as much the SVM's one. In [151] SVM and SVM-delayed scheme outperformed ANN in all real-time control performance metrics in the control of multiple myoelectric degrees of freedom. In the study reported in [167], instead, while SVM exceeded LDA in the accuracy of discrimination of two fNIRS-based mental tasks, LDA was faster in generating control commands.

Undoubtedly SVMs have several weaknesses: the choice of the kernel and of the hyperparameters may be extremely time-consuming. Moreover, the optimal design for multiclass SVM classifiers is still in question, whereas ANNs can have any number of outputs, thus easily solving such problem. Also, in large-scale tasks, the required quadratic programming of a standard SVM may require extensive memory and high algorithmic complexity and the big amount of support vectors may slow test phases.

The choice of the overall best classifier to detect physiological patterns is still questionable, but evidences coming from the literature support the idea that SVMs and all their implementations are among the most appropriate choice for HCI design due to their robustness and versatility.

## 6. Conclusion

SVMs result to be among the most versatile classifiers for pattern recognition. Due to the numerous SVM implementations available and to the possibility to create a virtually unlimited number of SVMs, by changing kernel and hyperparameters, SVMs allow investigating the most different scenarios. In the case of HCI driven by EEG and EMG, SVMs proved to be more accurate and efficient than other classifiers, such as LDA and ANNs, and furthermore resulted to be particularly suitable for online implementations. Anyway, in order to provide correct results and to allow other researchers to replicate studies, it is essential to report all the details of the adopted SVM strategy: kernel type, SVM implementation, multiclass strategy, hyperparameters setting methodology, performance evaluation. In such a view, this review paper provides some fundamental information to correctly design SVM-based systems.

## Acknowledgment

This paper was partially based on a work supported by the Italian Space Agency (ASI), contract #2013-081-R0, for which we would like to thank Prof Mariano Bizzarri, Dr Simona Zoffoli and Dr Francesca Ferranti. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

## Appendix. List of acronyms

Area under Roc curve (AUC)
Artificial neural networks (ANN)
Brain–computer interface (BCI)
Common spatial patterns (CSP)
Cross validation (CV)
Decision tree (DT)
Directed acyclic graphs (DAG)
ElectroCorticoGraphy (ECoG)
ElectroEncephaloGram (EEG)
ElectroMyoGram (EMG)
Event-related potentials (ERPs)
functional magnetic resonance imaging (fMRI)
functional near-infrared spectroscopy (fNIRS)
Fuzzy least-squares SVM (FLS-SVM)
Fuzzy SVM (FSVM)
Gaussian mixture models (GMM)
Genetic algorithms (GA)
Human–computer interaction (HCI)
Human-machine interface (HMI)
Immune feature weighted SVM (IFWSVM)
k-neirest neighbors (kNN)
Least-squares SVM (LS-SVM)
Leave-one-out (LOO)
Linear discriminant analysis (LDA)
Linear logistic classifier (LLC)
Logistic model tree (LMT)
MagnetoEncephaloGraphy (MEG)
Multi-layer perceptron (MLP)
Multiple Kernel learning SVM (MKL-SVM)
One-against-rest (OAR)
One-against-all (OAO)
One-against-one (OAO)
One-Vs-All (OVA)
One-Vs-One (OVO)
Particle swarm optimization (PSO)
Principal component analysis (PCA)
Quadratic discriminant analysis (QDA)
Radial basis function (RBF)
Rapid serial visual presentation (RSVP)
Rates of torque development (RTD)
Relevance vector machine (RVM)
Sensorimotor rhythms (SMR)
Simple logistic regression (SLR)
Slow-cortical potentials (SCP)
Spatially-weighted SVM (sw-SVM)
Steady-state visually evoked potentials (SSVEP)
Support vector machines (SVMs)
Target torque (TT)
Transductive SVM (TSVM)
$\varepsilon$-support vector regression ($\varepsilon$-SVR)

## References


[1] Olson G M and Olson J S 2003 Human–computer interaction: psychological aspects of the human use of computing *Annu. Rev. Psychol.* **54** 491–516

[2] Jaimes A and Sebe N 2007 Multimodal human–computer interaction: a survey *Comput. Vis. Image Underst.* **108** 116–34

[3] Beaudouin-Lafon M 1993 An overview of human–computer interaction *Biochimie* **75** 321–9

[4] Turk M 2014 Multimodal interaction: a review *Pattern Recognit. Lett.* **36** 189–95

[5] Holz E M, Botrel L, Kaufmann T and Kübler A 2015 Long-term independent brain–computer interface home use improves quality of life of a patient in the locked-in state: a case study *Arch. Phys. Med. Rehabil.* **96** S16–26

[6] Schettini F *et al* 2015 Assistive device with conventional, alternative, and brain–computer interface inputs to enhance interaction with the environment for people with amyotrophic lateral sclerosis: a feasibility and usability study *Arch. Phys. Med. Rehabil.* **96** S46–53

[7] Huang Z, Wang Z, Lv X, Zhou Y, Wang H and Zong S 2014 A novel functional electrical stimulation-control system for restoring motor function of post-stroke hemiplegic patients *Neural Regeneration Res.* **9** 2102–10

[8] Ortiz-Catalan M, Sander N, Kristoffersen M B, Håkansson B and Brånemark R 2014 Treatment of phantom limb pain (PLP) based on augmented reality and gaming controlled by myoelectric pattern recognition: a case study of a chronic PLP patient *Front. Neurosci.* **8** 1–7

[9] Kashihara K 2014 A brain–computer interface for potential non-verbal facial communication based on EEG signals related to specific emotions *Front. Neurosci.* **8** 1–12

[10] Carabalona R, Grossi F, Tessadri A, Castiglioni P, Caracciolo A and de Munari I 2012 Light on! Real world evaluation of a P300-based brain–computer interface (BCI) for environment control in a smart home *Ergonomics* **55** 552–63

[11] Khushaba R N, Kodagoda S, Liu D and Dissanayake G 2013 Muscle computer interfaces for driver distraction reduction *Comput. Methods Prog. Biomed.* **110** 137–49

[12] Tanaka A and Knapp R B 2002 Multimodal interaction in music using the electromyogram and relative position sensing *Presented at the Conf. on New Interfaces for Musical Expression (NIME '02)* pp 1–6

[13] Quitadamo L R, Marciani M G, Cardarilli G C and Bianchi L 2008 Describing different brain computer interface systems through a unique model: a UML implementation *Neuroinformatics* **6** 81–96

[14] Lotte F, Congedo M, Lécuyer A, Lamarche F and Arnaldi B 2007 A review of classification algorithms for EEG-based brain–computer interfaces *J. Neural Eng.* **4** R1–13

[15] Chowdhury R H, Reaz M B I, Ali M A B M, Bakar A A A, Chellappan K and Chang T G 2013 Surface electromyography signal processing and classification techniques *Sensors* **13** 12431–66

[16] Vapnik V N 1998 *Statistical Learning Theory* 1st edn (New York: Wiley-Interscience)

[17] Chang C-C and Lin C-J 2011 LIBSVM: a library for support vector machines *ACM Trans. Intell. Syst. Technol.* **2** 1–27

[18] Collobert R and Bengio S 2001 SVMTorch: support vector machines for large-scale regression problems *J. Mach. Learn. Res.* **1** 143–60

[19] Cherkassky V and Mulier F 2007 *Learning from Data: Concepts, Theory, and Methods* (Hoboken, NJ: Wiley)

[20] Cavrini F 2016 *Hand Gesture Recognition for the Benefit of Transradial Amputees MSc Thesis* La Sapienza University, Rome, Italy

[21] Webb A R 2003 *Statistical Pattern Recognition* (New York: Wiley)

[22] Bishop C 2007 *Pattern Recognition and Machine Learning* (New York: Springer)

[23] Theodoridis S and Koutroumbas K 2009 *Pattern Recognition* 4th edn (Boston, MA: Elsevier)

[24] Alpaydin E 2014 *Introduction to Machine Learning* 3rd edn (Cambridge, MA: MIT Press)

[25] Riillo F, Quitadamo L R, Cavrini F, Gruppioni E, Pinto C A, Pastò N C, Sbernini L, Albero L and Saggio G 2014 Optimization of EMG-based hand gesture recognition: supervised versus unsupervised data preprocessing on healthy subjects and transradial amputees *Biomed. Signal Process. Control* **14** 117–25

[26] Chang C-C and Lin C-J 2001 Training v-support vector classifiers: theory and algorithms *Neural Comput.* **13** 2119–47

[27] Chen P-H, Lin C-J and Schölkopf B 2005 A tutorial on $\nu$-support vector machines *Appl. Stoch. Models Bus. Ind.* **21** 111–36

[28] de Souza B F, de Carvalho A C P L F, Calvo R and Ishii R P 2006 Multiclass SVM model selection using particle swarm optimization *Presented at the 6th Int. Conf. on Hybrid Intelligent Systems* (doi: 10.1109/HIS.2006.264914)

[29] Lessmann S, Stahlbock R and Crone S F 2006 Genetic algorithms for support vector machine model selection *Presented at the Int. Joint Conf. on Neural Networks (IJCNN '06)* pp 3063–69

[30] Montañés E, Barranquero J, Díez J and del Coz J J 2013 Enhancing directed binary trees for multi-class classification *Inf. Sci.* **223** 42–55

[31] Suykens J A K and Vandewalle J 1999 Least squares support vector machine classifiers *Neural Process. Lett.* **9** 293–300

[32] Yang D-P, Zhao J-D, Gu Y-K, Wang X-Q, Li N, Jiang L, Liu H, Huang H and Zhao D-W 2009 An anthropomorphic robot hand developed based on underactuated mechanism and controlled by EMG signals *J. Bionic Eng.* **6** 255–63

[33] Lin C-F and Wang S-D 2002 Fuzzy support vector machines *IEEE Trans. Neural Netw.* **13** 464–71

[34] Tsujinishi D and Abe S 2003 Fuzzy least squares support vector machines *Presented at the Int. Joint Conf. on Neural Networks* pp 1599–604

[35] Gönen M and Alpaydın E 2011 Multiple kernel learning algorithms *J. Mach. Learn. Res.* **12** 2211–68

[36] Jrad N and Congedo M 2012 Identification of spatial and temporal features of EEG *Neurocomputing* **90** 66–71

[37] Yao Y, Liu Y, Yu Y, Xu H, Lv W, Li Z and Chen X 2013 K-SVM: an effective SVM algorithm based on K-means clustering *J. Comput.* **8** 2632–9

[38] Tipping M E 2001 Sparse Bayesian learning and the relevance vector machine *J. Mach. Learn. Res.* **1** 211–44

[39] Ding S, Yu J, Qi B and Huang H 2012 An overview on twin support vector machines *Artif. Intell. Rev.* **42** 245–52

[40] Lv Y and Gan Z 2014 Robust $\varepsilon$-support vector regression *Math. Problems Eng.* **2014** 373571

[41] Mitchell T M 1997 *Machine Learning* (New York: McGraw-Hill)

[42] Bianchi L, Quitadamo L R, Garreffa G, Cardarilli G C and Marciani M G 2007 Performances evaluation and optimization of brain computer interface systems in a copy spelling task *IEEE Trans. Neural Syst. Rehabil. Eng.* **15** 207–16

[43] Witten I H, Frank E and Hall M A 2011 *Data Mining: Practical Machine Learning Tools and Techniques* 3rd edn (Burlington, MA: Morgan Kaufmann)

[44] Thompson D E *et al* 2014 Performance measurement for brain–computer or brain-machine interfaces: a tutorial *J. Neural Eng.* **11** 035001

[45] Miller K J, Schalk G, Fetz E E, den Nijs M, Ojemann J G and Rao R P N 2010 Cortical activity during motor execution, motor imagery, and imagery-based online feedback *Proc. Natl Acad. Sci. USA* **107** 4430–5

[46] Wolpaw J R and McFarland D J 2004 Control of a two-dimensional movement signal by a noninvasive brain–computer interface in humans *Proc. Natl Acad. Sci. USA* **101** 17849–54

[47] Coyle S M, Ward T E and Markham C M 2007 Brain–computer interface using a simplified functional near-infrared spectroscopy system *J. Neural Eng.* **4** 219–26

[48] Zhang J, Sudre G, Li X, Wang W, Weber D J and Bagic A 2011 Clustering linear discriminant analysis for MEG-based brain computer interfaces *IEEE Trans. Neural Syst. Rehabil. Eng.* **19** 221–31

[49] Cohen O, Koppel M, Malach R and Friedman D 2014 Controlling an avatar by thought using real-time fMRI *J. Neural Eng.* **11** 035006

[50] Naseer N and Hong K-S 2015 fNIRS-based brain–computer interfaces: a review *Front. Human Neurosci.* **9** 172

[51] Hiremath S V, Chen W, Wang W, Foldes S, Yang Y, Tyler-Kabara E C, Collinger J L and Boninger M L 2015 Brain computer interface learning for systems based on electrocorticography and intracortical microelectrode arrays *Front. Integr. Neurosci.* **9** 40

[52] Sellers E W, Krusienski D J, McFarland D J, Vaughan T M and Wolpaw J R 2006 A P300 event-related potential brain–computer interface (BCI): the effects of matrix size and inter stimulus interval on performance *Biol. Psychol.* **73** 242–52

[53] Sellers E W, Kübler A and Donchin E 2006 Brain–computer interface research at the University of South Florida Cognitive Psychophysiology Laboratory: the P300 Speller *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 221–4

[54] Rebsamen B, Guan C, Zhang H, Wang C, Teo C, Ang M H and Burdet E 2010 A brain controlled wheelchair to navigate in familiar environments *IEEE Trans. Neural Syst. Rehabil. Eng.* **18** 590–8

[55] Citi L, Poli R, Cinel C and Sepulveda F 2008 P300-based BCI mouse with genetically-optimized analogue control *IEEE Trans. Neural Syst. Rehabil. Eng.* **16** 51–61

[56] Escolano C, Antelis J M and Minguez J 2012 A telepresence mobile robot controlled with a noninvasive brain–computer interface *IEEE Trans. Syst. Man Cybern.* B **42** 793–804

[57] Aloise F, Schettini F, Aricò P, Salinari S, Guger C, Rinsma J, Aiello M, Mattia D and Cincotti F 2011 Asynchronous P300-based brain–computer interface to control a virtual environment: initial tests on end users *Clin. EEG Neurosci.* **42** 219–24

[58] Yu T, Li Y, Long J and Gu Z 2012 Surfing the internet with a BCI mouse *J. Neural Eng.* **9** 036012

[59] Zickler C, Halder S, Kleih S C, Herbert C and Kübler A 2013 Brain painting: usability testing according to the user-centered design in end users with severe motor paralysis *Artif. Intell. Med.* **59** 99–110

[60] Schreuder M, Riccio A, Risetti M, Dähne S, Ramsay A, Williamson J, Mattia D and Tangermann M 2013 User-centered design in brain–computer interfaces—a case study *Artif. Intell. Med.* **59** 71–80

[61] Yuan H and He B 2014 Brain–computer interfaces using sensorimotor rhythms: current state and future perspectives *IEEE Trans. Biomed. Eng.* **61** 1425–35

[62] LaFleur K, Cassady K, Doud A, Shades K, Rogin E and He B 2013 Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface *J. Neural Eng.* **10** 046003

[63] Onose G *et al* 2012 On the feasibility of using motor imagery EEG-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up *Spinal Cord* **50** 599–608

[64] Cao L, Li J, Ji H and Jiang C 2014 A hybrid brain computer interface system based on the neurophysiological protocol and brain-actuated switch for wheelchair control *J. Neurosci. Methods* **229** 33–43

[65] Perdikis S, Leeb R, Williamson J, Ramsay A, Tavella M, Desideri L, Hoogerwerf E J, Al-Khodairy A, Murray-Smith R and Millán J D R 2014 Clinical evaluation of BrainTree, a motor imagery hybrid BCI speller *J. Neural Eng.* **11** 036003

[66] Middendorf M, McMillan G, Calhoun G and Jones K S 2000 Brain–computer interfaces based on the steady-state visual-evoked response *IEEE Trans. Rehabil. Eng.* **8** 211–4

[67] Yin E, Zhou Z, Jiang J, Yu Y and Hu D 2015 A dynamically optimized SSVEP brain–computer interface (BCI) speller *IEEE Trans. Biomed. Eng.* **62** 1447–56

[68] Singla R, Khosla A and Jha R 2014 Influence of stimuli colour in SSVEP-based BCI wheelchair control using support vector machines *J. Med. Eng. Technol.* **38** 125–34

[69] Bakardjian H, Tanaka T and Cichocki A 2011 Emotional faces boost up steady-state visual responses for brain–computer interface *Neuroreport* **22** 121–5

[70] Hinterberger T, Schmidt S, Neumann N, Mellinger J, Blankertz B, Curio G and Birbaumer N 2004 Brain–computer communication and slow cortical potentials *IEEE Trans. Biomed. Eng.* **51** 1011–8

[71] Yilmaz O, Birbaumer N and Ramos-Murguialday A 2014 Movement related slow cortical potentials in severely paralyzed chronic stroke patients *Front. Human Neurosci.* **8** 1–8

[72] Li Y, Guan C, Li H and Chin Z 2008 A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system *Pattern Recognit. Lett.* **29** 1285–94

[73] Rakotomamonjy A and Guigue V 2008 BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller *IEEE Trans. Biomed. Eng.* **55** 1147–54

[74] Gu Z, Yu Z, Shen Z and Li Y 2013 An online semi-supervised brain–computer interface *IEEE Trans. Biomed. Eng.* **60** 2614–23

[75] Salvaris M and Sepulveda F 2009 Visual modifications on the P300 speller BCI paradigm *J. Neural Eng.* **6** 046011

[76] Huang Y, Erdogmus D, Pavel M, Mathan S and Hild Ii K E 2011 A framework for rapid visual image search using single-trial brain evoked responses *Neurocomputing* **74** 2041–51

[77] Furdea A, Ruf C A, Halder S, De Massari D, Bogdan M, Rosenstiel W, Matuz T and Birbaumer N 2012 A new (semantic) reflexive brain–computer interface: in search for a suitable classifier *J. Neurosci. Methods* **203** 233–40

[78] Matsumoto M and Hori J 2014 Classification of silent speech using support vector machine and relevance vector machine *Appl. Soft Comput.* **20** 95–102

[79] Vallabhaneni A and He B 2004 Motor imagery task classification for brain computer interface applications using spatiotemporal principle component analysis *Neurol. Res.* **26** 282–7

[80] Bai O, Lin P, Vorbach S, Li J, Furlani S and Hallett M 2007 Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG *Clin. Neurophysiol.* **118** 2637–55

[81] Boostani R, Graimann B, Moradi M H and Pfurtscheller G 2007 A comparison approach toward finding the best feature and classifier in cue-based BCI *Med. Biol. Eng. Comput.* **45** 403–12

[82] Zhou S-M, Gan J Q and Sepulveda F 2008 Classifying mental tasks based on features of higher-order statistics from EEG signals in brain–computer interface *Inf. Sci.* **178** 1629–40

[83] Xu Q, Zhou H, Wang Y and Huang J 2009 Fuzzy support vector machine for classification of EEG signals using wavelet-based features *Med. Eng. Phys.* **31** 858–65

[84] Hsu W-Y 2013 Independent component analysis and multiresolution asymmetry ratio for brain–computer interface *Clin. EEG Neurosci.* **44** 105–11

[85] Yang B, Han Z, Zan P and Wang Q 2014 New KF-PP-SVM classification method for EEG in brain–computer interfaces *Bio-Med. Mater. Eng.* **24** 3665–73

[86] Hsu W-Y 2014 Improving classification accuracy of motor imagery EEG using genetic feature selection *Clin. EEG Neurosci.* **45** 163–8

[87] Coyle D 2009 Neural network based auto association and time-series prediction for biosignal processing in brain–computer interfaces *IEEE Comput. Intell. Mag.* **4** 47–59

[88] Lei X, Yang P and Yao D 2009 An empirical bayesian framework for brain–computer interfaces *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 521–9

[89] Siuly S and Li Y 2012 Improving the separability of motor imagery EEG signals using a cross correlation-based least square support vector machine for brain–computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **20** 526–38

[90] Yu X, Park S-M, Ko K-E and Sim K-B 2013 Discriminative power feature selection method for motor imagery EEG classification in brain computer interface systems *Int. J. Fuzzy Logic Intell. Syst.* **13** 12–8

[91] Zhou B, Wu X, Zhang L, Lv Z and Guo X 2014 Robust spatial filters on three-class motor imagery EEG data using independent component analysis *J. Biosci. Med.* **2** 43

[92] Soman S and Jayadeva N 2015 High performance EEG signal classification using classifiability and the Twin SVM *Appl. Soft Comput.* **30** 305–18

[93] She Q, Ma Y, Meng M and Luo Z 2015 Multiclass posterior probability twin SVM for motor imagery EEG classification *Comput. Intell. Neurosci.* **2015** 251945

[94] Siuly S and Li Y 2014 Discriminating the brain activities for brain–computer interface applications through the optimal allocation-based approach *Neural Comput. Appl.* **26** 799–811

[95] Zhang Y, Zhou G, Jin J, Wang X and Cichocki A 2015 Optimizing spatial patterns with sparse filter bands for motor-imagery based brain–computer interface *J. Neurosci. Methods* **255** 85–91

[96] Artusi X, Niazi I K, Lucas M F and Farina D 2011 Performance of a simulated adaptive BCI based on experimental classification of movement-related and error potentials *IEEE J. Emerg. Sel. Top. Circuits Syst.* **1** 480–8

[97] Tam W-K, Tong K-Y, Meng F and Gao S 2011 A minimal set of electrodes for motor imagery BCI to control an assistive device in chronic stroke subjects: a multi-session study *IEEE Trans. Neural Syst. Rehabil. Eng.* **19** 617–27

[98] Farina D, do Nascimento O F, Lucas M-F and Doncarli C 2007 Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters *J. Neurosci. Methods* **162** 357–63

[99] Gu Y, do Nascimento O F, Lucas M-F and Farina D 2009 Identification of task parameters from movement-related cortical potentials *Med. Biol. Eng. Comput.* **47** 1257–64

[100] Liao X, Yao D, Wu D and Li C 2007 Combining spatial filters for the classification of single-trial EEG in a finger movement task *IEEE Trans. Biomed. Eng.* **54** 821–31

[101] Hung C-I, Lee P-L, Wu Y-T, Chen L-F, Yeh T-C and Hsieh J-C 2005 Recognition of motor imagery electroencephalography using independent component analysis and machine classifiers *Ann. Biomed. Eng.* **33** 1053–70

[102] Hsu W-Y 2012 Enhanced active segment selection for single-trial EEG classification *Clin. EEG Neurosci.* **43** 87–96

[103] Liao K, Xiao R, Gonzalez J and Ding L 2014 Decoding individual finger movements from one hand using human EEG signals *PloS One* **9** e85192

[104] Hsu W-Y and Hu Y-P 2015 Artificial bee colony algorithm for single-trial electroencephalogram analysis *Clin. EEG Neurosci.* **46** 119–25

[105] Sonkin K M, Stankevich L A, Khomenko J G, Nagornova Z V and Shemyakina N V 2015 Development of electroencephalographic pattern classifiers for real and imaginary thumb and index finger movements of one hand *Artif. Intell. Med.* **63** 107–17

[106] Yang H, Guan C, Chua K S G, Chok S S, Wang C C, Soon P K, Tang C K Y and Ang K K 2014 Detection of motor imagery of swallow EEG signals based on the dual-tree complex wavelet transform and adaptive model selection *J. Neural Eng.* **11** 035016

[107] Gysels E, Renevey P and Celka P 2005 SVM-based recursive feature elimination to compare phase synchronization computed from broadband and narrowband EEG signals in brain–computer interfaces *Signal Process.* **85** 2178–89

[108] Sun S and Zhang C 2006 Adaptive feature extraction for EEG signal classification *Med. Biol. Eng. Comput.* **44** 931–5

[109] Liao X, Yao D and Li C 2007 Transductive SVM for reducing the training effort in BCI *J. Neural Eng.* **4** 246–54

[110] Li J, Zhang L, Tao D, Sun H and Zhao Q 2009 A prior neurophysiologic knowledge free tensor-based scheme for single trial EEG classification *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 107–15

[111] Cabrera A F, Farina D and Dremstrup K 2010 Comparison of feature selection and classification methods for a brain–computer interface driven by non-motor imagery *Med. Biol. Eng. Comput.* **48** 123–32

[112] Guo L, Wu Y, Zhao L, Cao T, Yan W and Shen X 2011 Classification of mental task from EEG signals using immune feature weighted support vector machines *IEEE Trans. Magn.* **47** 866–9

[113] Bhattacharyya S, Konar A and Tibarewala D N 2014 Motor imagery, P300 and error-related EEG-based robot arm movement control for rehabilitation purpose *Med. Biol. Eng. Comput.* **52** 1007–17

[114] Li X, Wang L and Sung E 2008 AdaBoost with SVM-based component classifiers *Eng. Appl. Artif. Intell.* **21** 785–95

[115] De Massari D, Pacheco D, Malekshahi R, Betella A, Verschure P F M J, Birbaumer N and Caria A 2014 Fast mental states decoding in mixed reality *Front. Behav. Neurosci.* **8** 415

[116] Li X, Chen X, Yan Y, Wei W and Wang Z J 2014 Classification of EEG signals using a multiple kernel learning support vector machine *Sensors* **14** 12784–802

[117] Bojorges-Valdez E, Echeverría J C and Yanez-Suarez O 2015 Evaluation of the continuous detection of mental calculation episodes as a BCI control input *Comput. Biol. Med.* **64** 155–62

[118] Peterson D A, Knight J N, Kirby M J, Anderson C W and Thaut M H 2005 Feature selection and blind source separation in an EEG-based brain–computer interface *EURASIP J. Adv. Signal Process.* **19** 3128–40

[119] Wang L, Zhang X, Zhong X and Zhang Y 2013 Analysis and classification of speech imagery EEG for BCI *Biomed. Signal Process. Control* **8** 901–8

[120] Liu Y-H, Wu C-T, Cheng W-T, Hsiao Y-T, Chen P-M and Teng J-T 2014 Emotion recognition from single-trial EEG based on kernel Fisher's emotion pattern and imbalanced quasiconformal kernel support vector machine *Sensors* **14** 13361–88

[121] Iacoviello D, Petracca A, Spezialetti M and Placidi G 2015 A real-time classification algorithm for EEG-based BCI driven by self-induced emotions *Comput. Methods Prog. Biomed.* **122** 293–303

[122] Kang J-S, Park U, Gonuguntla V, Veluvolu K C and Lee M 2015 Human implicit intent recognition based on the phase synchrony of EEG signals *Pattern Recognit. Lett.* **66** 144–52

[123] Qin J, Li Y and Sun W 2007 A semisupervised support vector machines algorithm for BCI systems *Comput. Intell. Neurosci.* **2007** 1–9

[124] Kayikcioglu T and Aydemir O 2010 A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data *Pattern Recognit. Lett.* **31** 1207–15

[125] Duan L, Ge H, Ma W and Miao J 2015 EEG feature selection method based on decision tree *Bio-Med. Mater. Eng.* **26** S1019–25

[126] Daley H, Englehart K, Hargrove L and Kuruganti U 2012 High density electromyography data of normally limbed and transradial amputee subjects for multifunction prosthetic control *J. Electromyogr. Kinesiol.* **22** 478–84

[127] Young A J, Kuiken T A and Hargrove L J 2014 Analysis of using EMG and mechanical sensors to enhance intent recognition in powered lower limb prostheses *J. Neural Eng.* **11** 056021

[128] Phinyomark A, Phukpattaranont P and Limsakul C 2011 A review of control methods for electric power wheelchairs based on electromyography signals with special emphasis on pattern recognition *IETE Tech. Rev.* **28** 316–26

[129] Hamedi M, Salleh S-H, Astaraki M and Noor A M 2013 EMG-based facial gesture recognition through versatile elliptic basis function neural network *Biomed. Eng. Online* **12** 73

[130] Ferreira A, Celeste W C, Cheein F A, Bastos-Filho T F, Sarcinelli-Filho M and Carelli R 2008 Human-machine interfaces based on EMG and EEG applied to robotic systems *J. Neuroeng. Rehabil.* **5** 10

[131] Fukuda O, Tsuji T, Kaneko M and Otsuka A 2003 A human-assisting manipulator teleoperated by EMG signals and arm motions *IEEE Trans. Robot. Autom.* **19** 210–22

[132] Christodoulou C I, Kaplanis P A, Murray V, Pattichis M S, Pattichis C S and Kyriakides T 2012 Multi-scale AM–FM analysis for the classification of surface electromyographic signals *Biomed. Signal Process. Control* **7** 265–9

[133] Güler N F and Koçer S 2005 Classification of EMG signals using PCA and FFT *J. Med. Syst.* **29** 241–50

[134] Fraser G D, Chan A D C, Green J R and MacIsaac D T 2014 Automated biosignal Quality analysis for electromyography using a one-class support vector machine *IEEE Trans. Instrum. Meas.* **63** 2919–30

[135] Stirling L M, von Tscharner V, Kugler P F and Nigg B M 2011 Classification of muscle activity based on effort level during constant pace running *J. Electromyogr. Kinesiol.* **21** 566–71

[136] Lucas M-F, Gaufriau A, Pascual S, Doncarli C and Farina D 2008 Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization *Biomed. Signal Process. Control* **3** 169–74

[137] Castellini C, Gruppioni E, Davalli A and Sandini G 2009 Fine detection of grasp force and posture by amputees via surface electromyography *J. Physiol.* **103** 255–62

[138] Tavakolan M, Xiao Z G and Menon C 2011 A preliminary investigation assessing the viability of classifying hand postures in seniors *Biomed. Eng. Online* **10** 79

[139] Chen X and Wang Z J 2013 Pattern recognition of number gestures based on a wireless surface EMG system *Biomed. Signal Process. Control* **8** 184–92

[140] Geethanjali P and Ray K K 2015 A low-cost real-time research platform for EMG pattern recognition-based prosthetic hand *IEEE/ASME Trans. Mechatronics* **20** 1948–55

[141] Liu J 2015 Adaptive myoelectric pattern recognition toward improved multifunctional prosthesis control *Med. Eng. Phys.* **37** 424–30

[142] Castellini C and van der Smagt P 2009 Surface EMG in advanced hand prosthetics *Biol. Cybern.* **100** 35–47

[143] Kawano S, Okumura D, Tamura H, Tanaka H and Tanno K 2009 Online learning method using support vector machine for surface-electromyogram recognition *Artif. Life Robot.* **13** 483–7

[144] Naik G R, Kumar D K and Jayadeva N 2010 Twin SVM for gesture classification using the surface electromyogram *IEEE Trans. Inf. Technol. Biomed.* **14** 301–8

[145] Liu Y-H, Huang H-P and Weng C-H 2007 Recognition of electromyographic signals using cascaded kernel learning machine *IEEE/ASME Trans. Mechatronics* **12** 253–64

[146] Oskoei M A and Hu H 2008 Support vector machine-based classification scheme for myoelectric control applied to upper limb *IEEE Trans. Biomed. Eng.* **55** 1956–65

[147] Shenoy P, Miller K J, Crawford B and Rao R N 2008 Online electromyographic control of a robotic prosthesis *IEEE Trans. Biomed. Eng.* **55** 1128–35

[148] Yan Z, You X, Chen J and Ye X 2009 Motion classification of EMG signals based on wavelet packet transform and LS-SVMs ensemble *Trans. Tianjin Univ.* **15** 300–7

[149] Khokhar Z O, Xiao Z G and Menon C 2010 Surface EMG pattern recognition for real-time control of a wrist exoskeleton *Biomed. Eng. Online* **9** 41

[150] Cesqui B, Tropea P, Micera S and Krebs H I 2013 EMG-based pattern recognition approach in post stroke robot-aided rehabilitation: a feasibility study *J. Neuroeng. Rehabil.* **10** 75

[151] Ameri A, Kamavuako E N, Scheme E J, Englehart K B and Parker P A 2014 Support vector regression for improved real-time, simultaneous myoelectric control *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 1198–209

[152] Shim H-M and Lee S 2015 Multi-channel electromyography pattern classification using deep belief networks for enhanced user experience *J. Cent. South Univ.* **22** 1801–8

[153] Tsai A-C, Luh J-J and Lin T-T 2015 A novel STFT-ranking feature of multi-channel EMG for motion pattern recognition *Expert Syst. Appl.* **42** 3327–41

[154] Xie H-B, Huang H, Wu J and Liu L 20015 A comparative study of surface EMG classification by fuzzy relevance vector machine and fuzzy support vector machine *Physiol. Meas.* **36** 191–206

[155] Huang H, Zhang F, Hargrove L J, Dou Z, Rogers D R and Englehart K B 2011 Continuous locomotion-mode identification for prosthetic legs based on neuromuscular-mechanical fusion *IEEE Trans. Biomed. Eng.* **58** 2867–75

[156] Miller J D, Beazer M S and Hahn M E 2013 Myoelectric walking mode classification for transtibial amputees *IEEE Trans. Biomed. Eng.* **60** 2745–50

[157] Tolambiya A, Thomas E, Chiovetto E, Berret B and Pozzo T 2011 An ensemble analysis of electromyographic activity during whole body pointing with the use of support vector machines *PloS One* **6** e20732

[158] Xu X, Zhang Y, Luo Y and Chen D 2013 Robust bio-signal based control of an intelligent wheelchair *Robotics* **2** 187–97

[159] Hamedi M, Salleh S-H and Noor A M 2015 Facial neuromuscular signal classification by means of least square support vector machine for MuCI *Appl. Soft Comput.* **30** 83–93

[160] Friedman J H 1989 Regularized discriminant analysis *J. Am. Stat. Assoc.* **84** 165–75

[161] Farrell T R and Weir R F 2007 The optimal controller delay for myoelectric prostheses *IEEE Trans. Neural Syst. Rehabil. Eng.* **15** 111–8

[162] Laskov P, Gehl C, Krüger S and Müller K-R 2006 Incremental support vector learning: analysis, implementation and applications *J. Mach. Learn. Res.* **7** 1909–36

[163] Bordes A, Ertekin S, Weston J and Bottou L 2005 Fast kernel classifiers with online and active learning *J. Mach. Learn. Res.* **6** 1579–619

[164] Kim T-K, Stenger B, Kittler J and Cipolla R 2010 Incremental linear discriminant analysis using sufficient spanning sets and its applications *Int. J. Comput. Vis.* **91** 216–32

[165] Polikar R, Upda L, Upda S S and Honavar V 2001 Learn ++: an incremental learning algorithm for supervised neural networks *IEEE Trans. Syst. Man Cybern.* C **31** 497–508

[166] Oweis R J and Hamdi N 2013 A comparison study on machine learning algorithms utilized in P300-based BCI *J. Health Med. Inform.* **4** 126

[167] Naseer N, Hong M J and Hong K-S 2013 Online binary decision decoding using functional near-infrared spectroscopy for the development of brain–computer interface *Exp. Brain Res.* **232** 555–64