Application of neural network and SVM to classify movement of rat in medical science

Farahnaz Sadoughi, Taha Samad Soltani, Mostafa Shanbe Zadeh

Department of Health Information Management, School of Health Management and Information Sciences, Tehran University of Medical Sciences, Tehran, Islamic Republic of Iran

Corresponding author email: t.ssoltany@razi.tums.ac.ir

ABSTRACT: Identifying mammalian movement states has recently become an important topic in biological science research. Accurate assessment and analysis of movement is a fundamental requirement. Rodents are often used as models in the sleep field due to their ready availability and the similarities of their movement to human movement. The goal of intra-cortical brain computer interface (BCI) is to restore the lost functionalities in disabled patients suffering from severely impaired movements. The project aim was to develop a decoding method based on a rat model. Previously recorded data and an already develop pre-processing method were used. The experimental design was developed starting from intra-cortical (IC) signal recorded in the rat primary motor cortex (M1). The data pre-processing included denoising with wavelet technique, spike detection, and feature extraction. After the firing rates of intra-cortical neurons were extracted, artificial neural network (ANN) and support vector machine (SVM) were applied to classify the rat movements into two possible classes, Hit or No Hit.

Key words: neural network, SVM, movement of rat, classification

INTRODUCTION

Movement is a physiological state composed of different stages. Electroencephalogram (EEG) analysis shows that typical patterns of activity are correlated with different stages of movement, wakefulness, and some path physiological processes, such as seizures. It is important for the researchers to be able to identify movement stages in some cases, for example, in movement deprivation and seizure studies (Carmena et al., 2003). Typically, movement stages can be identified by the combination of EEG, electromyogram (EMG), electrooculogram (EOG) and visual behavioral monitoring. However, it is a time-consuming task to score these vigilance states manually even when the analyzer is an expert. Consequently, reducing human intervention in this task is a significant topic in this field, and a variety of automated movement staging methods using different analyses have been developed over the past decades (Fraser et al. 2009). However, many of these methods still require human intervention. For example, a user may have to determine the appropriate parameters (thresholds) or participate in the entire classification procedure. As a result, selecting appropriate parameters for different experimental conditions is a very subjective task for researchers. To reduce human intervention and its influence as much as possible, it is necessary to develop a classification method that automatically builds the classifier using the collected data.

Traumatic lesions of the central nervous system (CNS), as well as neurodegenerative disorders, such as amyotrophic lateral sclerosis (ALS), brain stem stroke, muscular dystrophy, cerebralpalsy (CP) or "locked-in" syndrome are causes of severe motor deficits in a large number of patients. Every year, spinal cord alone is responsible for the occurrence of 11,000 new cases of paralysis in the United States alone. These cases have to be summed to the over 200,000 estimated patients who have to cope with partial (paraplegic) or almost total (i.e., quadriplegic) body paralysis only in the United States (Kennedy and Bakay,1998).

Considerable therapeutic interest is given to the option of restoring the voluntary motor controlling patients suffering from traumatic or degenerative lesions of the motor system. In fact, quadriplegic patients suffer from severe damage of the central nervous system that heavily limits their every-day life. These patients cannot move any of...
their limbs and muscles below the neck. For this reason any help, or form of communication with the external environment, can provide a big increment of their quality of life. Until very recently, most of the focus from the research on restoration of motor functions was directed to repair the damaged axons that mediate the communication and therefore the motor neurons or alpha motor neurons in the gray matter of spinal cord (Moran, 2010). Despite great effort it is still not possible to regenerate a large number of neurons to the original connection. This approach suggested that direct interfaces between subcortical motor centers and artificial actuators could bypass the spinal cord injuries allowing the patients to enact voluntary intentions. The challenge assumed that voluntary commands can be translated in real time from motor cortex to directly stimulate the musculature of the patients or an external prosthesis. In fact, although they cannot move, they can still think about movement. Recently there has been considerable progress in designing prosthesis to assist this sort of patients. The goal of designing brain computer interface (BCI) system is therefore to record these movement intentions, interpret them and use them to control an external device (Nicolelis and Lebedev, 2009).

Therefore, the main purpose of a brain computer interfaces (BCIs), also called brain machine interfaces (BMIs), is to interface the brain with an external device, such as a prosthesis (e.g. robotic arm), a wheelchair or a screen, to give to impair, partially paralyzed or completely paralyzed patient the chance to communicate or interact with the world (Nicolelis et al., 1998; Nicolelis, et al., 2001).

MATERIAL AND METHODS

In a real world application a generic BCI system is formed by a person (the user) controlling a device in an operating environment (e.g. wheelchair) through a series of functional components. A set of functional components between the user and the device is considered as the BCI interface technology. The BCI interface technology is developed to help a target population with specific ability to perform certain tasks with a device [21]. General Brain Computer Interface system (BCI) is composed by four major parts:

1. The data acquisition system, which records the neural activity from the brain. It consists in an interface (e.g. electrodes) that can be implanted in the cortex or externally in the scalp and the device that effectively records and stores the signals. The recordings can be invasive or non-invasive; the signal can be of different types. Depending on the nature of these recordings the BCI can be implemented for different applications.
2. A signal processing algorithm which analyses and interprets the neural signals as control commands. The signal processing part links the recordings to an effector.
3. The external device which is used as effector and controlled directly by the neural signals. It can be a visual signal (e.g computer cursor) or a complicated robotic or prosthetic system.
4. A feedback signal (e.g. visual or audio) send from the device to the user in order to improve the brain plasticity and the accuracy of the movement (Nobunaga et al. 1999).

![Functional components of a BCI system](#)
Problem formulation
To implement artificial neural network (ANN) and support vector machine (SVM) classifiers based on intracortical signal to detect forelimb hit related classes.

A BCI system is usually characterized from the type of output that it provides. Although the external device is a fundamental part of the BCI, it is quite difficult to provide a real-time control commands which is able to generate correct feedbacks. Such a closed loop system is too complex to be implemented in a short time. For this reason, the data collected were analyzed off line with a program implemented in MATLAB. This gave the opportunity to develop before a robust algorithm to be implemented and test in real time in future experiments. Moreover, it was possible to compare different types of classification and decide which model to implement in a real-time BCI.

Solution strategy
In order to investigate whether it is possible from neuronal activity, recorded from the primary motor cortex (M1), to detect different types of movement, a big amount of data needed to be analyzed and classified. The data were collected in a previous experiment and the data processing was performed to denoise the signal and extract the features of interest. In order to detect different classes of paw movement, neuronal firing rates were classified as related to the paw movement or not, in Hit and NoHit classes respectively.

Figure 2. Block diagram representing the steps necessary to design a BCI system to be tested in a rat model.

First part: Sprague-Dawely rats were trained to hit a retractable paddle, and then 16 channels electrodes were implanted for obtaining intra-cortical recordings. The pre-processing included data denoising, spike detection and features extraction.

Second part: Afterwards, the firing rates extracted from the related intra-cortical signals, were used as inputs data for two classification methods: artificial neural network (ANN) and support vector machine (SVM). The first question to be addressed was whether the denoising process was effective or not to achieve a lower classification error. To answer this question a comparison between the misclassification error rates were done before and after denoising for both the techniques. The second question was whether there were any differences between ANN and SVM classification results.

Figure 3. Block diagram representing the data processing and the classification.
RESULT AND DISCUSSION

Intra-cortical (IC) signals recorded from the rat motor cortex (M1) were used to decode the brain activity and correlate it with the forelimb movements of the rat. The firing rates of intracortical neurons were analyzed in order to be classifying into two possible classes: Hit and NoHit.

Behavioral training

Five adult male Sprague-Dawley rats (470 g § 30) were included in this research, according to the requirements of the "Danish Committee for the Ethical Use of Animals in Research". The rats were placed in an operant conditioning cage equipped with a retractable paddle lever and a food reward deliver mechanism via a pellet dispenser. The rats task were to hit the response paddle lever three times consecutively with a forepaw in order to obtain the food reward 4.1.

Three of them preferred to use the left paw, two preferred the right paw. The number of repetitions was chosen to be three to be sure that the rat was doing that specific task, and not a random combination of hits. Only the first hit out of three repetitions was consecutively analyzed, because it was free from previous hitting movement artifacts. On the contrary the second and the third hits could also be very near in time to each other, and for this reason be affected by previous hit movements’ artifacts. It was quite important for the movement classification to analyze only the hits corresponding to a correct paw movement.

The concept of successful hit was defined as:
1. The rat hit the response paddle lever three times consecutively
2. The three hits were performed in less than 6 s

The concept of wrong hit was defined as:
1. The rat hit before the paddle was ready (paddle lever out of the paddle case)
2. The rat used both paws to hit the paddle
3. The rat hit the paddle with the unexpected paw (according to the preference)

Trials corresponding to the definition of wrong hit were discarded from further processing.

Experimental setup

The experimental setup consisted of four main components: the rat cage with the paddle lever and the food reward mechanisms, the TDT system for data acquisition, a digital camera, and a computer. A digital pulse was sent to the recording system (TDT) once the rat hit the lever of the response paddle, to synchronize the neural data with the movement. These digital pulses were simultaneously recorded by a Lab view VI program via data acquisition card (NI USB-6259BNC, National instrument, USA) with two aims: counting the number of hits and control the status of the lever i.e. ready for hitting or not ready. After a set of three consecutive hits the paddle lever was automatically retracted for 9 s to allow the rat to eat the reward and to avoid the recording of wrong muscular activity (e.g., chewing). A 16 channels homemade tungsten microelectrode array was chronically implanted in the M1 area of the rat brain corresponding to the preferred paw.

Figure 4. Experimental setup scheme: the paddle lever was protracted from the paddle case and ready to be hit by the rat.
After the recording sessions an electrical stimulation was applied on the cortical electrode to assess the neural correspondence between the signal and the movement. A monophasic stimulus of 100 Hz, pulse width 200 \(\mu\)m and amplitude 0.1 to 1.5 MA was used. The rat was awake in a prone position with the paw not supported. The channels corresponding to the forepaw movement were considered as reliable or good channels, the other, corresponding to forepaw and neck movements, or no movement, were classified as not good channel.

The data analysis was carried out off-line using MATLAB (Math Works, Natick, USA). The blockdiagram in Figure 4 shows the data processing and the classification parts. The first part of the data processing, called pre-processing, consisted in: denoising the raw signal with wavelet techniques, spikes detection and features extraction, to group and count the spikes into time windows. The second part consisted in the classification realized with two methods: artificial neural network (ANN) and support vector machine (SVM).

**Time windows: Hit and NoHit data**

Each session contained a different number of trials that could vary depending on the number of wrong hits. The raw signals were composed by intra-cortical (IC) activity recorded from 16 different channels. For every session and every channel the first hit out of three consecutive hits, composing each trial and each channel, was considered in the data processing. For every first hit movement a time window of 500 ms was extracted, from -400 ms (before the first hit) to +100 ms (after the first hit), where \(t=0\) ms represents the first hit moment. The part of the signal which contained the information about the planning and programming of the forepaw movement was supposed to be before the movement itself, taking place at \(t=0\) ms. The execution of movement occurred after the motor command was processed. The data in this time window were referred as Hit data. A corresponding window with the same length, 500 ms, were extracted before the Hit data, from -900 ms to -400 ms, and was called NoHit data. This time window represented a transition time between walking, chewing etc. to hitting. For this reason it was an appropriate window for extracting the NoHit data, corresponding to normal brain activity.

**Feature extraction**

After the signal was denoised and the spikes were detected, the firing rate features related to the paw movement were extracted. The features extraction was important since it determined the input of the classifier, and therefore the level of discrimination between different classes. Two possible classes Hit and NoHit were chosen to classify the paw movement. These two classes corresponded to the correct classification for the features extracted from the Hit data window and the NoHit data window respectively.

Three features characterizing the Hit data set and three features characterizing the NoHit dataset were extracted from every trial. The features corresponded with the number of spikes counted in bins of 5 ms and summed into time window of 120 ms. The three time windows of 120 ms for the Hit data were chosen as following:

- Inside the Hit data [-400 ms +100 ms]
- First interval [-400 ms -280 ms]
- Second interval [-280 ms -160 ms]
- Third interval [-160 ms -40 ms]

The NoHit data:

- Inside the NoHit data [-900 ms -500 ms]
- First interval [-900 ms -780 ms]
- Second interval [-780 ms -660 ms]
- Third interval [-640 ms -540 ms]

Where \(t=0\) ms corresponds to the first hit time. Figure 5 provides a visual example.

Figure 5. The figure shows an example of IC raw signal from a single channel corresponding to a hit movement.
This features extraction is supported by a study of (Stark and Abeles, 2007) which reported that neurons connected with activity in the forelimb of the rat began to discharge 30 to 50 ms before forepaws contact. They discovered that the combined activity of forepaw movement preceded detectable lever movement by 150 ms, and EMG recordings by 100 ms. This evidence justified the time interval chosen for the features extraction.

**Artificial neural network design**

After the feature extraction, the classification part was developed. In order to design an artificial neural network (ANN), some choices were made, according to the ANN function. In particular, the main focus was on whether to use supervised or not supervised learning, how many feature to insert in the input layer, the number of hidden layers, the neurons into them and the type of output. About the data set partition it was crucial to decide the percentages of training, testing and validation data, and whether to employ a validation technique to achieve more robust results. Some choices were already made according to the recordings and to the solution strategy, such as the supervised learning, which was independent from the particular technique used but from the feature extraction step. The ANN internal parameters were chosen after a series of pilot experiments conducted in a particular session. This specific session was chosen because in the previous experiment it gave the better classification results. A supervised learning technique was used to classify the features, since the data set, formed by Hit and NoHit data, was paired with the correct classification labels. In particular, the data sets called Hit data set represented the features correlated with a correct movement of the paw. The data set called NoHit data set represented the features correlated with a normal brain activity. The following list presents the basic step to develop an ANN in MATLAB:

1. Collect the data
2. Extract the features of interest (firing rate)
3. Initialize:
   - Input vector: a vector containing the features characterizing the data coming from the two classes
   - Target vector: a vector containing the correct label for the classification
4. Build the network
5. Train the network
6. Test the network
7. Plot the result

In order to choose the correct number of inputs for the ANN, some pilot experiments were conducted on the features. Consequently also the number of neurons in the input layer were three. The output layer contained two neurons that assumed different state between 1 and -1. The choice between the two output neurons (the classes) was made considering the one with the highest value between -1 and 1. To divide the data for the pilot experiments the following partition was chosen: training (70%), validation (10%) and test (20%). For the purpose of the project, it was necessary to implement a neural network for pattern recognition. A feed forward network was therefore built, with one hidden layer and an output layer. The default input processing functions were removeconstantrows and mapminmax. For outputs, the default processing functions were also removeconstantrows and mapminmax. Mapminmax normalized inputs/targets to fall in the range [1, 1] and removeconstantrows removed inputs/targets that are constant (Stark and Abeles, 2007).

In order to use the information coming from all 16 channels of the electrode, or at least from the channels related to the movement of the rat forepaw, called good channels, more features were considered in the input vector. According with the number of good channels the number of features was increased by three for channel considered as "good". In the case of rat 4 thenumber of good channels available were five, therefore 3 features each channel, means that the number of features for the input layer were fifteen. The model presented fifteen features in the input layer, ten neurons in the hidden layer and two neurons in the output layer.

From these results it could be concluded that increasing the number of features the number of misclassified samples decreased. In fact, more features added more information to the classification process. Below is shown how the performance of the neural network behaves adding one channel per time, in a random run without averaging the results.
Each data set, one for every session, was divided into three sub groups:

Training set which was used to build and train the network, to compute the gradient and to update the network weights and biases. First, the classifier was trained with data for which the correct classification was known. Therefore the training set contained the correct answer.

Validation set which was used to validate the result. Also for this set the correct classification was known but was not used to change the classifier parameters. The validation error normally decreases during the initial phase of training, as does the training set error.

However, when the network begins to over fit the data, the error on the validation set typically began to rise. The network weights and biases are selected from the ANN at the minimum of the validation set error, this indicates a poor division of the data set.

The testing set which was used to test the trained network, compute the misclassification error rate and to compare different classification methods. In general, if the error on the test set reaches a minimum at a significantly different iteration number than the validation set error, this indicates a poor division of the data set.

The data set was divided into train (80%) and test (20%). The validation was done for every neural network inside the training phase, and therefore the data set to train the neural network was divided into training set and validation set. A standard function, that divides the input data randomly, was adopted. The neural network tool command dividerand was used. For both Hit and NoHit data sets, the following partition was adopted:
1. Train the NN 80%
2. Test the NN 20%

The training set contained 80% of data for training effectively and 20% of data for the validation. To avoid possible misleading results deriving from the narrow number of samples, and to make the result more robust, M-fold cross validation, with equal to 5 was applied to the data. The procedure was developed and applied to each session as following:
1. Divide randomly the number of trials in 5 equally sized groups
2. Use 5-1 groups to train the system and 1 to test
3. Do this 5 times and average the results

To evaluate the classifier performance in the testing set was used the misclassification error rate. The misclassification error rate, as describe in detail, was defined as the number of misclassified samples divided by the total number of samples in the testing set. This procedure was repeated 5 times, the misclassification error rate was computed for every one of the five testing set and averaged in the end. Before starting and after every loop in the 5 fold cross validation the network was reinitialized with the command.
Support vector machine classification

The support vector machine (SVM) classification algorithm has a different concept to build the decision boundary from the usual linear methods: it does not base the construction of boundaries (e.g., line, curves or hyperplane separating classes) in all data points but only in those that are considered critical being nearest to it. So the points that are nearest to the other class became more important and discriminant. The points closer to the boundary are called support vectors, and the support vector classifier try to maximize the distance between the critical support vectors to provide a better separation between classes. Finding the boundary that maximizes the margin is a classical optimization problem.

The same general concept for the classification used for the ANN is still valid in the support vector machine technique, which also performed a supervised learning. The same partition strategy was used for the SVM, giving the same features as input for the classification and the same target vector. Three features, corresponding with three time windows, for every channel were used like input. The number of inputs varied dealing with the session and according with the number of good channel found. In the target vector 0 was assigned as a label to the Hit data and 1 to the NoHit data.

To validate the results M fold cross validation was performed, with M equal to 5. The dataset was divided into 5 groups, the first time; four were used for training (80%) and one for testing (20%). In the second loop another combination of four was chosen and so on for five times. After the fifth loop, an average between the results was done, and the mean misclassification error was found.
Summary of the SVM design choices

- Type of learning: Supervised learning
- Number of features in the input vector: 3 features for each channel, multiplied by the number of good channels
- Data partition: training set (80%) and testing set (20%)
- Training: Sequential Minimal Optimization
- Transformation function: Polynomial kernel
- Validation: M-fold cross validation, with $M = 5$

Perfect classification never occurs; in fact a classification error can occur if classes overlap. There are several methods for quantifying the classification errors, among these can be found the confusion matrix and in particular misclassification error rate (Suminski, et al., 2009).

The confusion matrix was used to quantify the results of the classification. The confusion matrix gave the number of correct and incorrect samples classifications, usually as percentage, formed by 2 rows and 2 columns, which represented respectively the 'input', the true result from the classification, and the output, the result suggested by the classifier. The principal diagonal presented the data with a correct classification, on contrary, in the other diagonal two squares represented the misclassified data.

In a classification task there were two possible types of errors:
- Type I: incorrectly recognized samples, also called False Positive (FP), which were hits classified as no hits
- Type II: Not recognized sample, also called False Negative (FN), which were no hits classified as hits

True and false are referring to the correct or incorrect classification; positive and negative refer instead to the class to which the sample is assigned after the classification. Positive was assigned to the hit movements. To prove the efficiency of the denoising algorithms a comparison between classification of denoised features and features obtained without denoising is proposed for both methods. Afterwards, ANN and SVM misclassification error rates are compared to analyze the differences between the two classification methods.

Features extracted from denised and not denoised data were classified using ANN. The misclassification error rates were calculated in both cases. To evaluate the effect of the denoising, the two misclassification error rates were then compared. The misclassification error rate in the case of denoised data represented the minimum classification error rate obtained out of 100 different denoising combination of wavelets and thresholds. The misclassification error rate in the case of not denoised data was simply the misclassification error rate obtained from the features extracted from raw data. After the minimum misclassification error rates were extracted for each one of the 26 sessions, for the denoised and not denoised data, they were plotted together for each session. The t-test assesses whether the means of two groups are statistically different from each other. The p-value calculated for all the rats was lower than 0.05. It can be concluded that with a confidence of 95% the two groups represented statistically different populations. Therefore applying denoising technique made a difference in the results, providing a lower misclassification error rate for the ANN.

It was interesting to examine which combinations of wavelets and thresholds provided the better denoising. The minimum misclassification error rate was extracted for each session, out of 100 denoising combinations (wavelet and threshold).

Figure 9. The plots show in red the misclassification error rate obtained classifying features extracted from the not denoised data.
The same technique was used to determine if also with the support vector machine classifier the denoising algorithm produced an increasing in the classifier performance and therefore a decreasing of the misclassification error rate. The misclassification error rates obtained classifying features extracted from raw data (without denoising) were compared with the misclassification error rates obtained classifying the features extracted from denoised data in Figure 10.

Every point in the graph for the denoised data represents the minimum misclassification error rate extracted from 100 different denoising pairs (wavelet and threshold). Every point for the data without denoising represents instead the misclassification error rate computed for each session.

It can be observed that the denoised signal always provided a better feature for the classification. In fact the misclassification error rate referring to denoised features was in all cases lower. A paired t-test was performed to determine if the denoising was effective.

For every rat the minimum misclassification error computed from denoised data corresponded to the configuration of denoising (denoising pair) that allowed the result. A table similar to the previous one was obtained. The pairs that allowed a lower misclassification error rate were: (db2, 0.5), (coif2,0.8), (sym2,0.5) and (sym6,0.6).

![Figure 10](image)

Figure 10. The plots show in red the misclassification error rates obtained classifying features extracted from the not denoised data and in blue the misclassification error rates obtained classifying features extracted from denoised data, using in both cases SVM classifiers.

In this study no spikes sorting were performed. On the contrary, in some previous studies the spiking activity from single neurons was extracted. Furthermore, the decision to avoid the spike sorting is supported by (Suminski, et al., 2009), who reported that multi-units activity (MUA) provided an accurate prediction of the upcoming movement. MUA recordings were obtained more easily than single unit’s activity (SUA) and were stable over time. Compared with local field potential (LFP), MUA were less redundant, and compared with other intra-cortical signal they were more informative. In their articles (Suminski, et al., 2009) demonstrated also that predictions based on multichannel MUA were superior to those ones based on spikes and LFP. For this reason the choice to use multi-neuronal activity was not a simplification, but a way to investigate whether the information coming from an entire population could be decoded with the same, or better performance.

RESULTS

The results of the classification showed that the two methods, ANN and SVM, were comparable. This type of task was classified erroneously in average in the 14% of the cases (in the best case) and in the 38% of the cases
(in the worst case) with ANN, and in the 14\% and 39\% with SVM, respectively. Both of them performed non-linear model, capable of solving paradigms that linear computing cannot. The SVM was a classification algorithm fitting this problem because it allowed developing a robust classifier using only a few free parameters even for high-dimensional data. In fact, the evaluation of the decision function at the heart of the SVM is based on the evaluation of a subset of the training data (the support vectors). SVM were successfully applied. This accuracy was better than the one predicted by Bayesian methods. Neural networks are a popular method that generates complex decision boundaries.

CONCLUSION

Brain computer interfaces (BCI) system aims to restore basic motor functions or to assist partially paralyzed or completely paralyzed patients. Recent studies showed that motor parameters, such as limb trajectory and cognitive parameters, such as the goal of an action, can be predicted by decoding motor related intra cortical signals from the primary motor cortex. In fact, mathematical model can be used to translate intra-cortical (IC) signals into features related to particular hand movements. The aim of the project was to investigate different decoding algorithms in order to investigate whether IC signals can distinguish a specific hit movement from the normal brain activity during hitting task.

A Sprague-Dawley rat was trained to hit a retractable paddle lever in an operant conditioning cage for three times consecutively. An electrode was placed within the topographical representation of the paw movement in the motor cortex, verified by visual inspection and electrical stimulation. IC signals were obtain in 26 sessions while the rat was performing the hitting task. The signal was first denoised with 100 different combinations of wavelets and thresholds. The firing rate features were extracted after the spikes detection and used as inputs for two types of nonlinear classifier: artificial neural network and support vector machine.

The results verified the denoising effectiveness and allowed to find some denoising pairs (wavelet and threshold) which gave better classification results. The two methods provided good misclassification error rates ranging from 14\% to 38\% for ANN, and 18\% to 39\% for SVM. Applying paired t test resulted that there was not statistical differences between the two groups of errors. Anyway the ANN achieved the best mean misclassification error rate.

The rat model was an appropriate choice related to the problem formulation. In fact it allowed the detection of the hit movement in an on-off task. The classifier can be implemented in a real-world application that require positive or negative outputs response or an application that can detect the intention to move.

REFERENCE