Soft Comput
DOI 10.1007/s00500-015-1937-5

FOCUS

Adaptive learning with covariate shift-detection for motor imagery-based brain–computer interface

Haider Raza[1] · **Hubert Cecotti**[1] · **Yuhua Li**[2] · **Girijesh Prasad**[1]

Abstract A common assumption in traditional supervised learning is the similar probability distribution of data between the training phase and the testing/operating phase. When transitioning from the training to testing phase, a shift in the probability distribution of input data is known as a covariate shift. Covariate shifts commonly arise in a wide range of real-world systems such as electroencephalogram-based brain–computer interfaces (BCIs). In such systems, there is a necessity for continuous monitoring of the process behavior, and tracking the state of the covariate shifts to decide about initiating adaptation in a timely manner. This paper presents a covariate shift-detection and -adaptation methodology, and its application to motor imagery-based BCIs. A covariate shift-detection test based on an exponential weighted moving average model is used to detect the covariate shift in the features extracted from motor imagery-based brain responses. Following the covariate shift-detection test, the methodology initiates an adaptation by updating the classifier during the testing/operating phase. The usefulness of the proposed method is evaluated using real-world BCI datasets (i.e. BCI competition IV dataset 2A and 2B). The results show a statistically significant improvement in the classification accuracy of the BCI system over traditional learning and semi-supervised learning methods.

Keywords Adaptive learning · Brain–computer interfaces · Covariate shift-detection · Transductive learning

1 Introduction

In traditional machine learning techniques, data are assumed to be drawn from stationary distributions. While training a traditional supervised classifier, it is commonly assumed that the input data distribution in the training set and the testing set follows the same probability distribution (Grossberg 1988; Mitchell 1997; Kelly et al. 1999; Vapnik 1999; Duda et al. 2001; Bishop 2006). However, in real-world applications, processes are non-stationary and are often characterized by a shifting nature, as the data distribution may shift over time. With applications working in non-stationary environments (NSEs), the data distribution shifts over time; in general this may be due to thermal drift, ageing effects, and noise. The scenario where the training set and testing set follow different distributions but the conditional distribution remains unchanged is known as covariate shift (Sugiyama et al. 2007; Li et al. 2010). In most of the real-world applications, non-stationarity is quite common, especially with the systems interacting with the dynamic and evolving environments, e.g., data coming from electroencephalogram (EEG)-based brain–computer interfaces (BCIs), share price prediction in stock market, and wireless sensor networks. Achieving high

Communicated by D. Neagu.

✉ Haider Raza
raza-h@email.ulster.ac.uk

Hubert Cecotti
h.cecotti@ulster.ac.uk

Yuhua Li
y.li@salford.ac.uk

Girijesh Prasad
g.prasad@ulster.ac.uk

[1] School of Computing and Intelligent Systems, Intelligent Systems Research Centre, Ulster University, Londonderry, UK

[2] School of Computing, Science and Engineering, University of Salford, Manchester, UK

Published online: 28 November 2015

classification accuracy in a BCI is a particularly challenging task because the signals may be highly variable over time.

A BCI is an alternative communication's means, which allows a user to express his or her will without muscle exertion, provided that the brain signals are properly translated into computer commands (Wolpaw et al. 2002). With an EEG-based BCI that operates online in real-time non-stationary/changing environments, it is required to consider input features that are invariant to shifts of the data during long and across sessions, or learning approaches that are able to detect the changes that may repeat overtime, to update the classifier in a timely fashion. The non-stationarities in the EEG may be caused by various reasons such as changing user attention level, electrode placement, and user fatigue (Li et al. 2010; Blankertz et al. 2008; Raza et al. 2015b). Due to these non-stationarities, it is expected to find notable variations or shifts in the EEG signals during trial-to-trial, and session-to-session transfers (Blankertz et al. 2002; Li et al. 2010; Arvaneh et al. 2013a; Raza et al. 2013a, 2015b). These variations often appear as covariate shifts in the EEG signals, wherein the input data distributions differ significantly between training/calibration and testing/operating phases, while the conditional distribution remains the same (Raza et al. 2013b; Satti et al. 2010; Sugiyama et al. 2007; Shimodaira 2000; Raza et al. 2014). To date, the low classification accuracy has been one of the main concerns of the developed BCI systems based on a motor imagery (MI) detection, which directly affects the reliability of the BCI (Li et al. 2010; Blankertz et al. 2008; Rezaei et al. 2006). To enhance the performance of BCI systems, several feature extraction, feature selection, and feature classification techniques have been proposed in the literature (Shahid and Prasad 2011; Suk and Lee 2013; Kuncheva and Faithfull 2014; Buttfield et al. 2006; Vidaurre et al. 2006; Coyle et al. 2009; Ramoser et al. 2000; Arvaneh et al. 2013a, b). A large variety of features have been used in MI-based BCI such as band powers, power spectral density, time frequency features, and common special patterns (CSP)-based features (Raza et al. 2015a). However, due to brain's non-stationary characteristics, the spatial distribution of the brain-evoked responses may change over time, resulting in shifts in feature distributions (Herman et al. 2008).

The main drawback of the solutions proposed in the related literature is the requirement of labeled data before starting the adaptation in the evaluation/operating phase (Li et al. 2010; Sugiyama 2012). Additionally, most of the shift-detection methods present in the literature are based on the batch processing for a dataset shift detection (Gama and Kosina 2014; Alippi et al. 2013; Elwell and Polikar 2011; Gama et al. 2014), so there is a time delay in shift-detection. Hence, for real-time systems, the batch processing methods are not beneficial where initiating adaptation in the nick-of-time is of supreme interest. In this paper, we present a novel design methodology for an adaptive classification, which monitors the covariate shift in the input streaming data (i.e., EEG features) through an exponential weighted moving average (EWMA) model-based covariate shift-detection (CSD) test (Raza et al. 2013a, b). The CSD test operates in two stages: the first stage deals with covariate shift-detection, and the second stage corresponds to the covariate shift validation. This two-stage structure helps in reducing the false detection rate, which may reduce an unnecessary retraining of the classifier. The classifier adaptation is only initiated once the covariate shift is confirmed using validation; after validation, the classifier is retrained based on the updated knowledge base (KB) discussed later in Sect. 4. The proposed method uses two different adaptation mechanisms to update the knowledge base (KB i.e., training data) of the classifier on the new knowledge. In the first method, a transductive learning approach is used to add the relevant information to the KB after each CSD. Moreover, the transductive learning is only used to increase the size of KB, but the overall classification is performed using an inductive classifier. In the second method, the KB is updated incrementally using the correctly predicted labels after each CSD. The experiments on the real-world datasets are used to show that the covariate shift can be adapted using the proposed method. Using the data from the BCI competition-IV 2A and -2B, we have demonstrated that the proposed method can outperform a traditional learning approach and other competing methods. It is to be noted that a preliminary work related to the proposed methodology was presented in our conference paper (Raza et al. 2014) and here, we extend the study of adaptive learning with covariate shift-detection by conducting an extensive experimental evaluation on motor imagery-based BCI datasets. In particular, our main focus is to account for covariate shift which may arise during session-to-session transfer in BCI experiments. In addition, we perform a thorough analysis on the feature extraction techniques, to extract better discriminative features for the classifier. The novel contributions of the paper can thus be summarized as follows:

- A covariate shift-adaptation model is introduced to address the effects of non-stationarity in the EEG signals.
- An EWMA-based CSD test is applied to detect the non-stationary changes in the principal component analysis (PCA)-based features of the motor imagery-based brain responses.
- Third, the proposed model updates its classification decision boundary online without making any a priori assumption about the distribution for the upcoming test data.

The remainder of the paper proceeds as follows: first, Sect. 2 describes the proposed methodology for the covariate shift-detection, -validation, and -adaptation; Sect. 3 presents an

application of the method to BCI. Then, the results are detailed in Sect. 4. Finally, the implications of the results are discussed in Sect. 5.

2 Methods

2.1 Adaptive learning problem formulation

Let us consider a learning framework in which training dataset is denoted by $X_{\mathrm{Tr}} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of observations, and a target label y_i is associated with each input x_i. Depending upon the number of inputs and outputs, x_i and y_i may be scalar or vector variables. In the following work, the training dataset is represented as initial KB. Let us consider a two-class classification problem, i.e., $y \in \{C_1, C_2\}$, where $y_i = C_1$, if x_i belongs to class ω_1, and $y_i = C_2$, if x_i belongs to class ω_2. For example, in support vector machine (SVM), we have $C_1 = -1$, and $C_2 = +1$. The probability distribution of the inputs at time i can thus be defined as $P(x_i) = P(\omega_1)P(x_i|\omega_1) + P(\omega_2)P(x_i|\omega_2)$, where $P(\omega_1)$, $P(\omega_2)$ are the prior probabilities of getting a sample of the classes ω_1 and ω_2, respectively, while $P(x_i|\omega_1)$, $P(x_i|\omega_2)$ are the conditional probability distribution for the time period i.

The goal is to predict the labels of upcoming samples $(\widehat{y_i})$ resulting in $X_{Ts} = \{(\widehat{y_i}|x_i)\}_{i=1}^{M}$, where M is the number of observations in the testing phase.

2.2 Algorithm overview

The proposed algorithm with the covariate shift-detection (CSD) belongs to the category of incremental learning (Elwell and Polikar 2011), where the learning model is updated at each CSD. The covariate shift monitoring is performed using the CSD-EWMA test (Raza et al. 2013a, b, 2015b). An advantage of using the CSD test is the enhanced accuracy in terms of low false-positives and low false-negatives. The proposed algorithm is a single classifier-based non-stationary learning (NSL) algorithm that uses the CSD-EWMA test for initiating adaptive corrective action. The algorithm is provided with a time-series training dataset KB, where KB $= X_{\mathrm{Tr}}$, and a classifier $\mathcal{F}$ is trained. In the evaluation phase, the CSD-EWMA test is used to monitor and detect the covariate shift. Then, the classifier $\mathcal{F}$ is used to classify the upcoming input data X_{Ts}.

The key elements of the proposed solution are

- CSD: CSD test monitors the stationarity of x_i, disregarding their supervised labels.
- $\mathcal{F}$: The pattern classifier $\mathcal{F}$ is used to classify the input samples.
- KB: The current knowledge base (KB) updated on each CSD.

The proposed solution is described in Algorithm 1. After a preliminary configuration phase of the initial classifier $\mathcal{F}$ and CSD on KB, the CSD is used to assess the process stationarity. As soon as the CSD-EWMA test detects a covariate shift in the upcoming unlabeled data, the classifier learned model becomes obsolete and has to be replaced with a newly configured/retrained model. At each CSD, the new information (i.e., $\mathrm{KB_{New}}$) becomes available containing the information about the new data distribution. Next, the $\mathrm{KB_{New}}$ is merged with existing KB, and a new KB is prepared. To prepare the updated KB, two methods are identified: the first is a transductive learning with CSD (TLCSD), and second is an adaptive learning with CSD (ALCSD). The interactions between the covariate shift-detection, -validation, and -adaptation stages are more clearly illustrated with the help of Figs. 1 and 2, which are explained in the following subsections.

Algorithm 1: Learning with CSD

1. Configure the classifier $\mathcal{F}$ based on the initial knowledge base KB= X_{Tr};
2. Configure the parameters λ and L for the CSD test using the KB;
3. **FOR** $i = 1$ to length(X_{Ts})
4. Receive new data x_i;
5. **IF** (CSD detects and validates a non-stationarity at time i), **THEN**
6. KB ← KB ∪ $\mathrm{KB_{New}}$
7. Retrain and adapt the classifier $\mathcal{F}$ on KB
8. **END**
9. Classify the input x_i by the classifier $\mathcal{F}$ and get the predicted label $\hat{y}_i$;
10. **END**

2.3 Covariate shift-detection (CSD)

The first step required in a CSD test is to detect the covariate shift in the process, possibly without relying on the prior information about the process data distribution before and after the shift. This is a crucial step for reconfiguring the classifier, and it acts as an alarm. The first stage of the test provides an initial estimate of the shift (i.e., where the actual shift has occurred). The first stage test is performed by an SD-

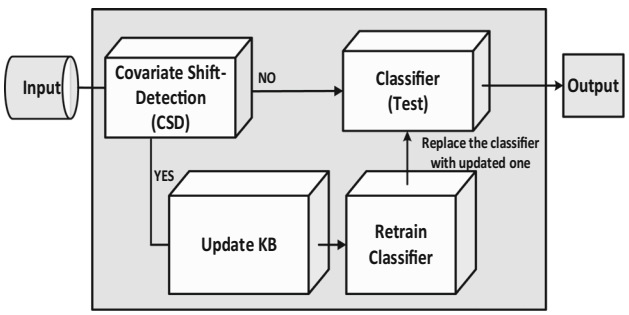


Fig. 1 Architecture of the adaptive learning design methodology

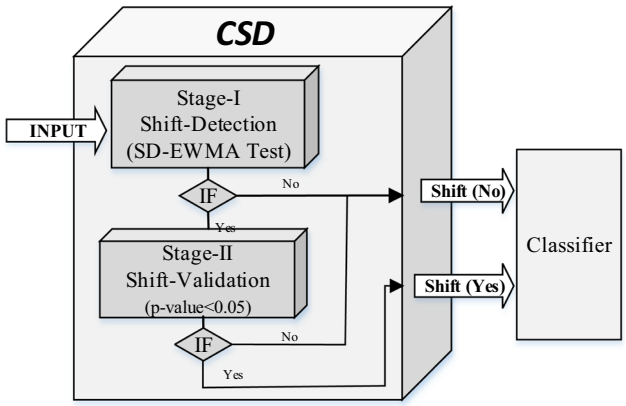


Fig. 2 A two-stage covariate shift-detection (CSD). Stage-I is for shift-detection and stage-II works for validation

EWMA test (Raza et al. 2013a). If the test outcome at the first stage is positive, then the second stage test gets activated, and a validation is performed to reduce the number of false alarms (Raza et al. 2013b). The second stage test/validation procedure is discussed in next sub-section. The choice of the smoothing constant λ and a control limit multiplier (L) are the important issue in the EWMA-CSD test. The choice of λ and L are discussed in Sect. 4.

In an EEG-based BCI, the EEG signals are obtained from multiple electrodes, and the application of a feature extraction procedure results in a set of features, and hence BCI input data are multivariate. Monitoring of such input processes independently may be misleading, e.g., if the probability that a variable exceeds three-sigma control limits is 0.0027, then a false detection rate of 0.27 % is expected. However, the joint probability that d variables exceed their control limits simultaneously is $(0.0027)^d$. So, the use of d-independent control-charts may provide highly distorted outcomes. A principal component analysis (PCA) is, therefore, used to reduce the dimensionality of the data (Rosenstiel et al. 2012; Kuncheva and Faithfull 2014). It provides fewer components, containing most of the variability in the data. We have used a single component to monitor the shift in the process using SD-EWMA test (Raza et al. 2013a) at the first stage.

2.4 Covariate shift-validation

According to Algorithm 1, the KB of the classifier has to be updated at each CSD. However, false positives (i.e., detection that does not correspond to a true shift in the input distribution) result in an unnecessary retraining. To counter this, we have introduced a covariate shift-validation procedure as part of a two-stage structure test (Raza et al. 2013b). This strategy aims at guaranteeing that the classifier relies on an up-to-date KB, and the classifier is only retrained on the occurrence of a valid shift. The covariate shift-validation procedure exploits two sets of observations generated before and at the CSD time point. The observations from the KB are assumed to be in its stationary state and are compared with data from the current trial, at the CSD time point. To validate the CSD from the stage-I, a multivariate Hotelling's T square statistical hypothesis test is used (Hotelling 1947). If the p value of the test is below 0.05, then the CSD is confirmed; otherwise it is considered as a false-alarm. On each CSD, the $\mathrm{KB}_{\mathrm{New}}$ is obtained based on the current shift in the data.

2.5 Covariate shift-adaptation

Once the CSD is validated, the adaptation phase starts (see Fig. 2). To adapt to the shift, re-training the classifier is required. In order to retrain the classifier, an additional set of input target pairs is necessary to prepare the KB. To get the set of input target pairs, we have investigated two ways for the KB management. In the first scenario (i.e., TLCSD), we have applied a transductive-inductive learning model to adapt to a potential covariate shift. However, the transduction part is only used to add new trials into $\mathrm{KB}_{\mathrm{New}}$, and an inductive classifier is used to classify the upcoming samples from the evaluation phase. The transduction part will only start once the covariate shift is detected and validated. In the second scenario (i.e., ALCSD), it is assumed that during the evaluation phase, a true label is available after each trial. Once the covariate shift is detected, then only correctly predicted labels are added into $\mathrm{KB}_{\mathrm{New}}$, the classifier is re-trained, and the updated classifier is used for further classification. This approach is similar to co-training (Zhu 2008) used in a semi-supervised learning (SSL), where the predicted labels are used to train another classifier.

Both the methods mentioned above that are used to adapt the classifier in relation to the covariate shift are presented hereafter.

2.5.1 Transductive learning with CSD (TLCSD)

A TLCSD model is based on a probabilistic K-nearest neighbor (KNN) method. Initially, according to Algorithm 1, at step 1, an inductive classifier $\mathcal{F}$ is trained on the initial KB, and at step 2, the parameters λ and L are set for the CSD test. Once the classifier $\mathcal{F}$ is trained, then an evaluation phase starts. At step 3, the parameters λ, L, $\mathrm{CR}_{\mathrm{Thres}}$, and K are set, wherein $\mathrm{CR}_{\mathrm{Thres}}$ is a confidence ratio threshold that is used to decide the usefulness of the trial, and K is the number of neighbors for the transductive learning. In the evaluation phase, the classifier takes the features as the input obtained from the testing data. The classifier initiates adaptation through transduction after every CSD. Each time the classifier initiates adaptation at step 7, it is considered as one epoch, and it takes Δm data points to predict the labels

through a transductive function $\mathcal{T}$, where Δm is the number of points between two shift-detection points, or from the start of evaluation phase to the first detection point. Once the adaptation is initiated at each epoch, the Euclidean distance $(d_{p,q})$ from the unlabeled data point x_p to the labeled data point x_q is computed as given below:

$$d_{(p,q)} = \|x_p - x_q\| \tag{1}$$

This provides a vector $\mathbf{D} = [d_{(p,q_1)}, \ldots, d_{(p,q_N)}]$ of Euclidean distances from unlabeled data point to the N number of labeled data points. Then, the K nearest neighbors are selected. For each of the K nearest points, an RBF kernel is used to compute the weight, as given in Eq. (2).

$$K(p,q) = \exp\left(-\frac{\|x_p - x_q^2\|}{2\sigma^2}\right) \tag{2}$$

From Eq. (2), we have $0 \leq K(p,q) \leq 1$. A weight with a high value implies the data-point's closeness to the unlabeled current feature. Thus, the weight for each neighbor is given by

$$R(i) = K(p, q_i) \tag{3}$$

Using $R(i)$ and the existing KB, for each of the classes a confidence ratios CR_{ω_i} is obtained by

$$\mathrm{CR}_{\omega_1} = P(\omega_1|x) = \frac{\sum_{i=1}^{K} R(i) * (y(i) == \omega_1)}{\sum_{i=1}^{K} R(i)} \tag{4a}$$

$$\mathrm{CR}_{\omega_2} = P(\omega_2|x) = \frac{\sum_{i=1}^{K} R(i) * (y(i) == \omega_2)}{\sum_{i=1}^{K} R(i)} \tag{4b}$$

The confidence ratio CR_{ω_i} attained from Eq. (4a) and (4b) may be viewed as a posterior probability of the class membership of the current unlabeled data point, as $\mathrm{CR}_{\omega_1} + \mathrm{CR}_{\omega_2} = 1$. This CR_{ω_i} acts as a belief or confidence, which determines if a data sample belongs to a particular class. In this step, for each observation from the Δm data points are obtained, and CR_{ω_i} to decide if both the trial's features and the estimated output labels should be added to the existing knowledge-base, i.e. if $\max(\mathrm{CR}_{\omega_1}, \mathrm{CR}_{\omega_2}) > \mathrm{CR}_{\mathrm{Thres}}$, then the couple (EEG signal corresponding to the trial, estimated output label) is added into $\mathrm{KB}_{\mathrm{New}}$; otherwise it is discarded. At step 7, this $\mathrm{KB}_{\mathrm{New}}$ is then merged into the existing KB. Based on the updated KB, the inductive classifier function is updated, and a new classifier $\mathcal{F}$ is obtained at step 8. Every time a new $\mathrm{KB}_{\mathrm{New}}$ is created, the classifier $\mathcal{F}$ is updated, and this process is repeated until all the M points in the testing phase are classified.

2.5.2 Adaptive learning with CSD (ALCSD)

In ALCSD, initially at step 1 of Algorithm 1, an inductive classifier $\mathcal{F}$ is trained with the initial KB of N labeled trials. Using KB at step 2, the parameter λ is obtained for the CSD test, and the control limit (L) for the CSD is set to $L = 2$. Then, an evaluation phase starts at step 4, and unlabeled features from X_{Ts} are processed sequentially for classification. At step 6, the CSD test is used to monitor the covariate shift. Once the covariate-shift is detected, it acts as an alarm to update the classifier. To update the classifier, new knowledge from the data is required. In order to obtain $\mathrm{KB}_{\mathrm{New}}$, it is assumed that in each trial, the true label is available, and among all predicted labels only correctly predicted labels through an inductive classifier are added into $\mathrm{KB}_{\mathrm{New}}$. KB is updated with the content of $\mathrm{KB}_{\mathrm{New}}$ at step 7. KB is used to retrain the classifier at step 8, and further at step 10, this updated classifier is used to classify the upcoming data. On each CSD, KB gets updated, and a new classifier is created.

3 Application to brain–computer interface

3.1 Data description

3.1.1 BCI Competition IV dataset 2A

The BCI competition IV dataset 2A (Tangermann et al. 2012) is comprised of the EEG data collected from nine subjects, namely (A01–A09), that were recorded during two sessions on separate days for each subject. The data consist of 25 channels, which include 22 EEG channels, and 3 monopolar EOG channels. Among the 22 EEG channels, 10 channels are selected for this study, which are responsible for capturing most of the motor imagery activities. The selected channels are presented in Fig. 3a. The data were collected on four different motor imagery tasks: left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4). Each session consists of six runs separated by short breaks, and each run comprised of 48 trials (12 for each class). The total numbers of 288 trials are in each session. Only the class 1 and the class 2 for left hand and right hand were considered in this study (i.e., 144 trials). For more details about the dataset kindly refer to (Tangermann et al. 2012). The motor imagery data from the session-I were used to train the classifiers, and the motor imagery data from the session-II were used as the test dataset.

3.1.2 BCI competition IV dataset 2B

BCI competition 2008-Graz dataset B (Tangermann et al. 2012) is a dataset consisting of EEG data from 9 subjects, namely (B01–B09). Three channel bipolar recordings (C3,

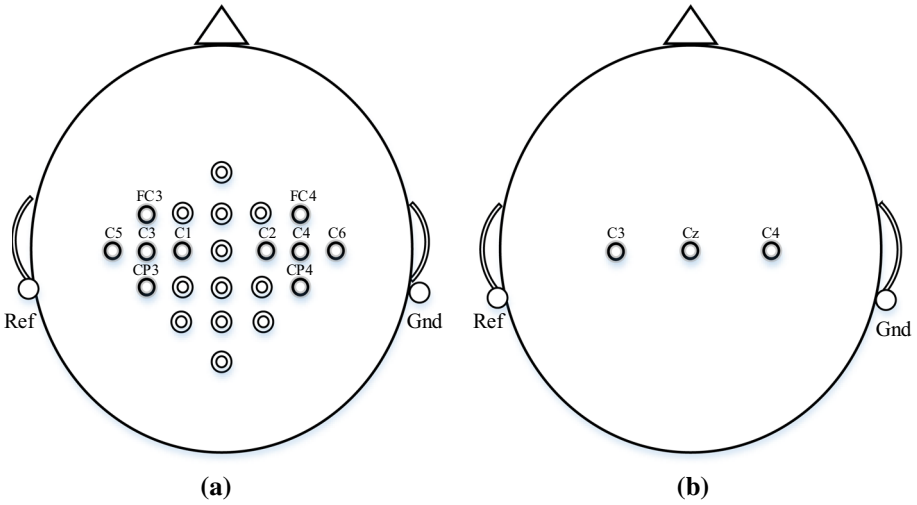


Fig. 3 Electrode montage corresponding to the international 10–20 system: **a** dataset 2A, among all 22 EEG channels, only ten channels are selected as shown in *black filled hollow circles*. **b** Dataset 2B, all channels are selected

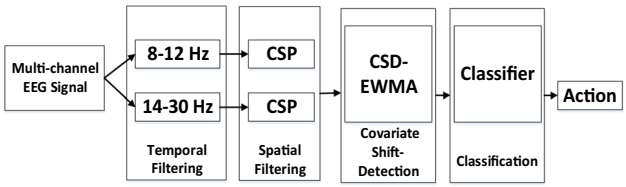


Fig. 4 Block diagram for the MI-based BCI. It consists of following five stages: initially multi-channel EEG signals are acquired, next the band-pass filtering is performed, and then the CSP features are obtained, and the covariate shift is monitored, and then features are classified using a pattern classifier. Finally, the BCI commanded action is performed

Cz, and C4) were acquired with a sampling frequency of 250 Hz; the montage is depicted in Fig. 3b. All signals were recorded monopolarly with the left mastoid serving as a reference and the right mastoid as a ground. For each subject, five sessions are provided. The motor imagery data from session-I and -II were used to train the classifiers, the data from session-III were used to obtain the hyperparameters (i.e., K and $\mathrm{CR}_{\mathrm{Thres}}$), and the motor imagery data from session-IV and -V were used to evaluate the performance of the test. Session-IV and -V consist of 160 trials each. Each trial is a complete paradigm of 8 s; for more details refer to Tangermann et al. (2012).

3.2 Data processing and feature extraction

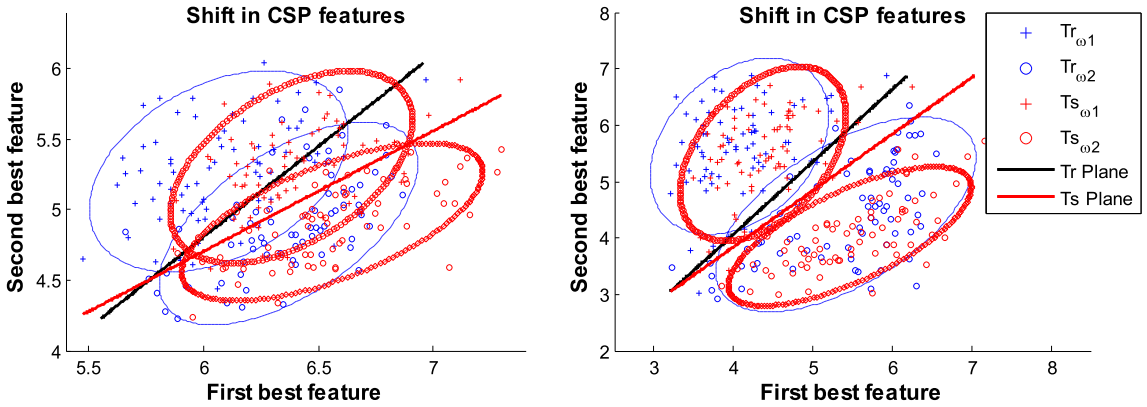
3.2.1 Temporal filtering

The second stage of the MI-based BCI block diagram (see Fig. 4) employs two filters that decompose the EEG signals into two different frequency bands. Two band-pass filters are used, namely (8–12) Hz (μ band) and (14–30) Hz (β band). These frequency ranges are used because they cover a stable frequency response related to MI-associated phenomena of event-related synchronization and de-synchronization (ERS/ERD). In the next sections, we consider a time segment of 3 s after the cue onsets for both data sets.

3.2.2 Spatial filtering

The third stage employs a spatial filter that maximizes the variance of spatially filtered signals under one condition, while minimizing it for the other condition. Raw EEG scalp potentials are known to have poor spatial resolution due to volume conduction. If the signal of interest is weak while other sources produce strong signals in the same frequency range, then it is difficult to classify two classes of EEG measurements (Blankertz et al. 2008). The neurophysiological background of motor-imagery based BCIs is that motor activity, both actual and imagined, causes an attenuation or increase of localized neural rhythmic activity, called event-related desynchronization (ERD) or event-related synchronization (ERS). The common-spatial-pattern (CSP) algorithm is highly successful in calculating spatial filters for detecting (ERD/ERS) (Ang et al. 2008, 2012). The objective of the CSP algorithm is to compute features whose variances are optimal for discriminating two classes of brain-evoked responses in EEG signal.

A pair of band-pass and spatial filters in the first and second stages perform spatial filtering of EEG signals that have been band-pass filtered in a specific frequency range. Thus, each pair of band-pass and spatial filter computes the CSP features that are specific to the band-pass frequency range. CSP is a technique to analyze multichannel data based on the recording from two classes (Blankertz et al. 2008). It is a data-driven supervised decomposition of signals parameterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{C\times C}$ (C: number of selected channels) that projects the single trial EEG signal $\boldsymbol{E} \in \mathbb{R}^{C}$ in the original sensor space to $\boldsymbol{Z} \in \mathbb{R}^{C}$, which lives in the surrogate sensor space, as follows:



(a) [8-12] Hz **(b)** [14-30] Hz

Fig. 5 Covariate shift in the EEG dataset 2A-subject A03, between training and testing input distribution for different frequency bands. **a** Mu band (8–12) Hz, and **b** beta band (14–30) Hz. The *red circles* denote the features of the *left hand* motor imagery, and *blue crosses* denote the features of the *right hand* motor imagery. The *black* and *red lines* represent the decision boundaries obtained by the training data and test data, respectively

$$\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{E}, \tag{5}$$

where $\boldsymbol{E} \in \mathbb{R}^{C\times T}$ is a EEG measurement data of single trial, C is the number of channels; T is the number of samples per channel. W is the CSP projection matrix. The rows of $\mathbf{W}$ are the spatial filters, and the columns of W are the common spatial patterns. The spatial filtered signal $\mathbf{Z}$ given in Eq. (5) maximizes the difference in the variance of the two classes of EEG measurements. A CSP analysis is applied to obtain an effective discrimination of mental states that are characterized by ERD/ERS effects. However, the variances of only a small number (m) of the spatial filtered signal are generally used as feature for classification. The m first and last rows of $\mathbf{Z}$, i.e. $\boldsymbol{Z_t}$, $t \in \{1 \ldots 2m\}$ form the feature vector $\boldsymbol{x_t}$ given by

$$\boldsymbol{x_t} = \log\left(\frac{\text{var}(\boldsymbol{Z_t})}{\sum_{i=1}^{2m}\text{var}(\boldsymbol{Z_i})}\right) \tag{6}$$

Here, $m = 1$. The CSP features from both frequency bands are combined to form the input features for training a classifier. Figure 5 shows the covariate shift in the CSP features for both training and test datasets for subject A03 over two different frequency bands mu (μ) (8–12) Hz and beta (β) (14–30) Hz. The blue crosses and red circles denote the features of the left hand and right hand motor imagery, respectively. The black line and red line represent the separation planes between the features of two classes obtained from two frequency bands as training and testing features, respectively. The separation planes are plotted for illustration purpose only.

3.3 Covariate shift-detection (CSD)

The fourth stage uses the CSD test on the CSP features. In both datasets, the data are generated from multiple channels, and for each channel two features are produced from each frequency band. To use CSD-EMWA, PCA is used to reduce the number of the features, and a single component is used to detect the covariate shift. To execute the CSD test, the smoothing constant λ is selected for each subject based on minimizing the sum of squares of 1-step-ahead prediction error method, and the control limit multiplier is set to $L = 2$. The choice of L has a major impact on the performance of the CSD test, a small value of L makes it more sensitive in detecting minor shifts in the data. The CSD test in the operational stage detects the shifts and validates it through its two-stage structure. If the CSD test is positive then a classifier is retrained on the KB.

3.4 Experimental setup and classification evaluation metrics

In order to evaluate the performance of the system, we have considered the classification accuracy (in %) as the measure of index. The experiments are performed using a linear support vector machine (SVM) pattern classifier $\mathcal{F}$. In CSD tests, the percentage (%) of covariate shift-detected and shift-validated is computed as given below:

$$\%\text{ of shift detected/validated} = \left(\frac{(\#\text{ shift detected/validated})}{\text{Total number of trials}}\right) \times 100 \tag{7}$$

The hyperparameters K and CR_{Thres} are required to be carefully selected. Two variants of the proposed learning method, namely $TLCSD_1$ and $TLCSD_2$, are, therefore, presented. In $TLCSD_1$, the hyperparameters are selected based on grid search to maximize the mean accuracy across subjects, with $K \in \{6, 12, 18\}$, and CR in the range (0.50–1). In $TLCSD_2$, the hyperparameters are determined for each subject, based on a grid search to maximize the accuracy of each subject (subject-dependent). In dataset 2A, session-I is divided into two parts; the first 80 % is used for training the pattern classifier while the remaining 20 % is used to determine the hyperparameters. The evaluation is then performed on the data from session-II. In dataset 2B, sessions I and II (240 trials) are used for training the pattern classifier, session III (160 trials) is used to obtain the hyperparameters, and sessions IV and V (320 trials) are used to evaluate the performance of the classifier. For each dataset, the accuracy corresponding to a tenfold cross-validation (10-CV) on the training data is provided. Moreover, the two variants for the proposed methods are evaluated and compared with a baseline method and a label propagation-based semi-supervised learning (SSL) algorithm. An upper bound (UB) is also provided. It is obtained by training the classifier ($\mathcal{F}$) on both the training and the test datasets, with an evaluation on the test data. The baseline method uses an inductive learning classifier with CSP features (Ramoser et al. 2000), but it does not adapt/re-train its pattern classifier over time. A graph-based SSL label propagation method (Zhu and Ghahramani 2002) has been considered for comparisons. To compare classifier performance with the baseline method, a two-sided Wilcoxon signed rank test is used to assess the statistical significance of the pairwise comparison at a confidence level of 0.05.

4 Results

4.1 Results for dataset 2A

The results corresponding to the choice of the smoothing constant λ and the CSD are presented in Table 1. The value of λ is obtained by minimizing the sum of squares of 1-step-ahead prediction errors. In the data of subject A05, a shift was detected 15 times (i.e. 10.42 % CSD), whereas it was detected only 7 times for subject A03 (i.e. 4.86 % CSD). For subject A05, the CSD decreased from 10.42 to 4.17 % after the covariate shift-validation stage, and for subject A03, the CSD decreased from 4.86 to 1.39 %. The validation stage thus helps to decrease the rate of false positives at stage-II; consequently the effort of unnecessary retraining the classifier is also reduced.

The classification accuracies on dataset 2A, for the different methods and for each subject, are given in Table 2. The

Table 1 Results for shift-detection and validation dataset 2A

Subject	Lambda	Shift-detected	Shift-validated
A01	0.10	7.64	2.78
A02	0.80	7.64	6.25
A03	1	4.86	1.39
A04	1	7.64	4.17
A05	0.30	10.42	4.17
A06	0.10	9.72	3.47
A07	0.10	8.33	6.25
A08	0.20	7.64	3.47
A09	0.50	6.94	2.78

10-CV average classification accuracy on the training dataset is 80.32 ± 10.25 %, where subject A08 is having a maximum accuracy of 93.57 %. For the baseline results, an inductive classifier is used for the classification on the test data without any adaptation on the CSP features. The baseline method gives an average accuracy of 73.46 ± 15.94 %, and subject A03, who has the less number of shifts, has the highest accuracy (92.36 %). The SSL label propagation method gives an average accuracy of 69.91 ± 18.22 %, which is inferior to the baseline method. In $TLCSD_1$, the parameters K and CR_{Thres} have been set to $K = 18$ and $CR_{Thres} = 0.70$, and the classification accuracy has improved slightly from 73.46 ± 15.94 to 74.07 ± 15.21 %.

For $TLCSD_2$, all the subjects have shown an improvement, except for subject A08. The average accuracy of $TLCSD_2$ is 74.92 ± 15.43 %, which represents a significant improvement compared to the baseline method (p value $= 0.0126$). In ALCSD, the results have shown a minor improvement in the performance against the baseline method with the mean accuracy of 73.84 ± 15.93 %; only subjects A01, A02, A03, and A07 have shown improvement. The accuracy of UB is 76.70 ± 15.33 %, and it represents the performance that can be achieved if all the data are available for training, showing that the knowledge of the test data points in the evaluation of the classifier can improve the performance by only 3.23 %.

4.2 Results for dataset 2B

The results for the choice of λ and the CSD are presented in Table 3. In this dataset, sessions IV and V are used for evaluation phase; hence for each session the CSD test is performed independently. In session IV, the subject B01 has the maximum number of CSD (10 %), and subject B04 has minimum number of CSD (1.88 %). After the covariate shift validation stage, the number of CSD decreased from 10 to 4.38 % for subject A01, and the number of CSD decreased from 1.88 to 0.63 % for subject A04. Moreover, in session V, subjects B06 and B08 have the maximum number of CSD (10 %),

Table 2 Classification accuracy (%) results from BCI competition IV-dataset 2A

	10-CV Tr	Baseline Eval	SSL Eval	$TLCSD_1$ Eval	$TLCSD_2$ Eval	ALCSD Eval	UB Eval
A01	85.71	89.58	79.17	90.28	90.28	90.28	90.28
A02	75.71	53.47	54.17	57.64	57.64	54.17	58.33
A03	92.86	92.36	93.06	93.06	95.14	93.75	97.22
A04	77.86	64.58	68.06	65.28	65.97	64.58	67.36
A05	61.43	59.03	45.14	59.72	61.11	57.64	59.03
A06	71.43	65.28	56.94	65.28	65.28	65.28	65.97
A07	84.29	59.72	54.17	59.72	61.11	62.50	70.83
A08	93.57	91.67	90.97	90.28	91.67	90.97	90.97
A09	80.00	85.42	87.50	85.42	86.11	85.42	90.28
Mean	80.32	73.46	69.91	74.07	74.92	73.84	76.70
Std	10.25	15.94	18.22	15.21	15.43	15.93	15.33
$^*p$ value			0.3047	0.2813	0.0156	0.5313	0.0156

* A two-sided Wilcoxon signed rank test is used to assess the statistical significance of the improvement at a confidence level of 0.05, the p value denotes the Wilcoxon signed rank test

Table 3 Results for shift-detection and validation dataset 2B

Subject	Lambda	Session IV		Session V	
		Shift-detected (%)	Shift-validated (%)	Shift-detected (%)	Shift-validated (%)
B01	0.10	10.00	4.38	6.88	3.13
B02	0.80	6.88	1.25	9.38	5.63
B03	1	6.88	2.50	8.13	6.25
B04	1	1.88	0.63	3.75	1.25
B05	0.30	7.50	4.38	6.88	3.75
B06	0.10	8.13	5.00	10.00	6.88
B07	0.10	6.25	5.63	7.50	2.50
B08	0.20	6.25	4.38	10.00	5.00
B09	0.50	8.13	4.38	8.13	3.75

and subject B04 has the minimum number of CSD (3.75 %). After the covariate shift-validation stage, the number of CSD decreased from 10 to 6.88 % for subject B06, and the number of CSD decreased from 10 to 5 % for subject B08.

The classification accuracies on dataset 2B, for the different methods and for each subject, are given in Table 4. The average accuracy with 10-CV is 70.71 ± 10.78 %, with subject B04 obtaining the maximum performance of 88.85 %. The baseline method gives 65.23 ± 13.98 % of average accuracy and subject B04 has the maximum accuracy of 93.13 %. The SSL-based label propagation method gives 62.74 ± 11.89 % average accuracy, which is below the baseline method accuracy. In $TLCSD_1$, the parameters K and CR_{Thres} have been fixed to $K = 18$ and $CR_{Thres} = 0.70$, and the classification accuracy has slightly improved from 65.23 ± 13.98 to 66.15 ± 13.64 %. Next, for $TLCSD_2$, all the subjects have shown an improvement. The average accuracy for $TLCSD_2$ is 69.72 ± 14.05 %, being statistically significantly better (p value $= 0.00039$) than the baseline method. In ALCSD, the results have shown a considerable improvement in the performance against the baseline method with the mean accuracy of 67.88 ± 14.16 %, which is statistically significantly better than the baseline method (p value $= 0.0039$). Moreover, for ALCSD, all the subjects have shown an improvement. The UB method reaches an accuracy of 73.33 ± 14.67 %. Figure 6, presents the average classification accuracy across subjects for both databases (2A and 2B).

5 Discussion

The proposed TLCSD and ALCSD methods for the EEG-based BCI are based on a covariate shift-detection and an adaptation framework. An EWMA-CSD test is used to detect the covariate shift. Once the shift is detected, an appropriate adaptive action is initiated to address the effect of the covariate shift. In TLCSD, the new information/knowledge obtained through transduction is used to update the KB (i.e., training data) of the inductive classifier. However, the main classification function is still inductive because the transduc-

Table 4 Classification accuracy (%) results from BCI competition IV-dataset 2B

	10-CV Tr	Baseline Eval	SSL Eval	$TLCSD_1$ Eval	$TLCSD_2$ Eval	ALCSD Eval	UB Eval
B01	70.42	69.69	66.56	69.06	70.31	71.88	75.00
B02	61.25	49.58	51.56	50.00	50.63	50.00	51.56
B03	56.67	51.56	49.38	48.44	52.81	52.81	52.19
B04	88.85	93.13	85.63	93.44	93.75	93.44	96.56
B05	76.15	52.81	51.25	62.81	63.75	54.37	77.19
B06	70.71	72.81	67.50	72.19	74.06	73.13	74.06
B07	84.29	58.13	56.25	59.38	61.88	62.50	70.00
B08	61.79	65.63	64.38	65.63	83.13	77.81	88.44
B09	66.25	73.75	72.19	74.38	77.19	75.00	75.00
Mean	70.71	65.23	62.74	66.15	69.72	67.88	73.33
Std	10.78	13.98	11.89	13.64	14.05	14.16	14.67
*p value			0.0391	0.6719	0.0039	0.0039	0.0039

* A two-sided Wilcoxon signed rank test is used to assess the statistical significance of the improvement at a confidence level of 0.05, the p value denotes the Wilcoxon signed rank test

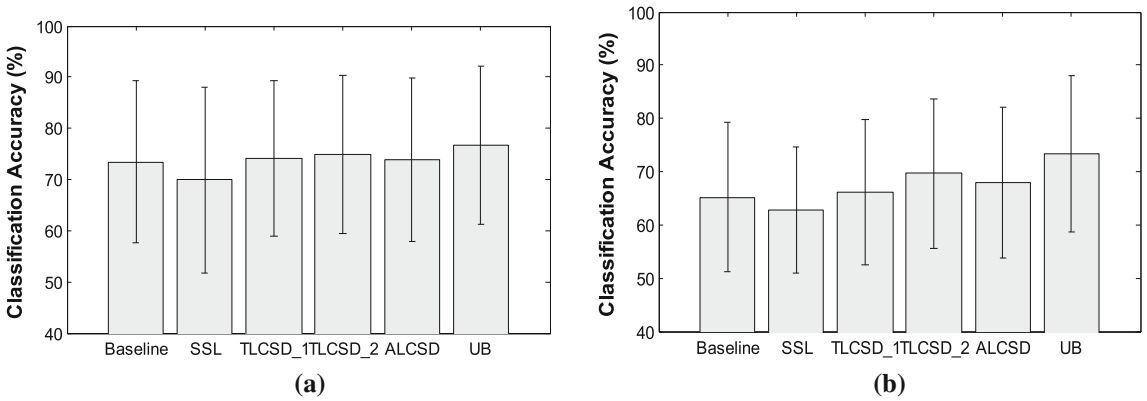


Fig. 6 Comparison of the mean accuracies for the proposed methods against the baseline, SSL, and UB on **a** the dataset 2A and **b** dataset 2B. The *box plot* represent the standard deviation across subjects

tive knowledge is only used to add more information into KB.

An important issue in the CSD is the choice of the control limit multiplier L. Considering small limit $L = 2$ means focusing on minor shifts, such as muscular artifacts arising during trial-to-trial transfer. However, the long-term non-stationarities may be accounted for by considering a large value of $L = 3$, such as during session-to-session transfer or run-to-run transfer. We have selected a small value of control limit multiplier $L = 2$, as our aim is to detect the covariate shift that arises during trial-to-trial transfers. The proposed learning techniques make use of CSD to detect the shift and then adapt to non-stationarities in the streaming EEG.

The parameter CR_{Thres} is used to decide whether the information in hand is useful or not. If the information is useful then it is added to the existing KB. The discarded information may come from a different distribution or it may have not provided much confidence to add into KB. The value of CR_{Thres} and K are important and are required to be carefully selected to achieve superior performance. For instance, for the method $TLCSD_1$, the value of CR_{Thres} is empirically selected in the range (0.50–1). In $TLCSD_2$, the parameters are selected based upon a grid search method and the accuracy is superior for both of the datasets. This implies that the performance of the proposed method depends upon the optimal choice of CR_{Thres}.

The experimental results demonstrated the effectiveness of the proposed covariate shift-detection and adaptation learning strategy. The results showed that the proposed method with CSP filters and optimized parameters is significantly better than the traditional learning methods and SSL with CSP filters. The combination of EWMA-based covariate shift-detection and adaptive learning is thus a good choice for learning in non-stationary environments. The robustness of the CSD test plays an important role in initiating a correct adaptive action.

6 Conclusion

The proposed methodology is a flexible tool for adaptive learning in non-stationary environments and effectively accounts for the effect of the covariate shifts. In this paper, two methods (TLCSD and ALCSD) were proposed for the covariate shift-adaptation using a two-stage covariate shift-detection test. The CSD test in the first stage uses the SD-EWMA test; and in the second stage, the multivariate Hotelling's T square statistical hypothesis test is used. The CSD test is found very effective in detecting the covariate shifts in the data in real-time. Based on the detected significant shifts, the algorithm initiates adaptive corrective action. The performance of the proposed methods was evaluated on multivariate cognitive task detection problem in the EEG-based BCIs simulated with BCI competition IV datasets 2A and 2B, and a superior classification accuracy was obtained as both TLCSD and ALCSD have shown statistically significant improvement. This work is planned to be extended further by employing the CSD into the task of fault monitoring.

Acknowledgments H. R. was supported by Ulster University Vice-Chancellor's research scholarship (VCRS). G. P. and H. C. were supported by the Northern Ireland Functional Brain Mapping Facility project (1303/101154803), funded by InvestNI and the Ulster University. G. P. and H. R. were also supported by the UKIERI DST Thematic Partnership project "A BCI operated hand exoskeleton based neuro-rehabilitation system" (UKIERI-DST-2013-14/126).

Compliance with ethical standards

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

References

Alippi C, Boracchi G, Roveri M (2013) Just-in-time classifiers for recurrent concepts. IEEE Trans Neural Networks Learn Syst 24:620–634. doi:10.1109/TNNLS.2013.2239309

Ang KK, Chin ZY, Zhang H, Guan C (2008) Filter bank common spatial pattern (FBCSP). In: Proceedings of the international joint conference on neural networks (IJCNN), pp 2390–2397

Ang KK, Chin ZY, Wang C et al (2012) Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. Front Neurosci 6:39. doi:10.3389/fnins.2012.00039

Arvaneh M, Guan C, Ang KK, Quek C (2013a) Optimizing spatial filters by minimizing within-class dissimilarities in electroencephalogram-based brain–computer interface. IEEE Trans Neural Netw Learn Syst 24:610–619. doi:10.1109/TNNLS.2013.2239310

Arvaneh M, Guan C, Quek C (2013b) EEG data space adaptation to reduce intersession nonstationary in brain–computer interface. J Neural Comput 25:1–26. doi:10.1162/NECO_a_00474

Bishop CM (2006) Pattern recognition and machine learning. Springer, New York

Blankertz B, Curio G, Müller K-R (2002) Classifying single trial EEG: towards brain computer interfacing. In: Advances in neural information processing systems, pp 157–164

Blankertz B, Tomioka R, Lemm S et al (2008) Optimizing spatial filters for robust EEG single-trial analysis. IEEE Signal Process Mag 25:41–56

Buttfield A, Ferrez PW, Millán JDR (2006) Towards a robust BCI: error potentials and online learning. IEEE Trans Neural Syst Rehabil Eng 14:164–168. doi:10.1109/TNSRE.2006.875555

Coyle D, Prasad G, McGinnity TM (2009) Faster self-organizing fuzzy neural network training and a hyperparameter analysis for a brain–computer interface. IEEE Trans Syst Man Cybern 39:1458–1471

Duda RO, Hart PE, Stork DG (2001) Pattern recognition. Wiley-Interscience, New York, USA

Elwell R, Polikar R (2011) Incremental learning of concept drift in nonstationary environments. IEEE Trans Neural Netw 22:1517–1531. doi:10.1109/TNN.2011.2160459

Gama J, Kosina P (2014) Recurrent concepts in data streams classification. Knowl Inf Syst 40(3):489–507. doi:10.1007/s10115-013-0654-6

Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv 4(1):1–44

Grossberg S (1988) Nonlinear neural networks: principles, mechanisms, and architectures. Neural Netw 1:17–61. doi:10.1016/0893-6080(88)90021-4

Herman P, Prasad G, McGinnity TM (2008) Designing a robust type-2 fuzzy logic classifier for non-stationary systems with application in Brain-computer interfacing. In: Proceedings of IEEE international conference on systems, man and cybernetics (SMC 2008), Singapore, 2008

Hotelling H (1947) Multivariate quality control-illsutrated by the air testing of sample bombsights. In: Techniques of statistical analysis, Chap II. pp 111–184

Kelly M, Hand D, Adams N (1999) The impact of changing populations on classifier performance. In: Proceedings of the fifth ACM SIGKDD. ACM, pp 367–371

Kuncheva L, Faithfull W (2014) PCA feature extraction for change detection in multidimensional unlabeled data. IEEE Trans Neural Netw Learn Syst 25:69–80

Li Y, Kambara H, Koike Y, Sugiyama M (2010) Application of covariate shift adaptation techniques in brain–computer interfaces. IEEE Trans Biomed Eng 57:1318–1324. doi:10.1109/TBME.2009.2039997

Mitchell T (1997) Machine learning. McGraw Hill, Boston, USA

Ramoser H, Müller-Gerking J, Pfurtscheller G (2000) Optimal spatial filtering of single trial EEG during imagined hand movement. IEEE Trans Rehabil Eng 8:441–446. doi:10.1109/86.895946

Raza H, Prasad G, Li Y (2013a) Dataset shift detection in non-stationary environments using EWMA charts. In: Proceedings—2013 IEEE international conference on systems, man, and cybernetics, SMC 2013, pp 3151–3156

Raza H, Prasad G, Li Y (2013b) EWMA based two-stage dataset shift-detection in non-stationary environments. In: IFIP Advances in information and communication technology. Springer, Berlin, Heidelberg, pp 625–635

Raza H, Prasad G, Li Y (2014) Adaptive learning with covariate shift-detection for non-stationary environments. In: 14th UK workshop on computational intelligence (UKCI), 2014. IEEE. Bradford, pp 1–8

Raza H, Cecotti H, Prasad G (2015a) Optimising frequency band selection with forward-addition and backward-elimination algorithms

in EEG-based brain-computer interfaces. In: International joint conference on neural networks (IJCNN), pp 1–7. doi:10.1109/IJCNN.2015.7280737

Raza H, Prasad G, Li Y (2015b) EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments. Pattern Recognit 48:659–669. doi:10.1016/j.patcog.2014.07.028

Rezaei S, Tavakolian K, Nasrabadi AM, Setarehdan SK (2006) Different classification techniques considering brain computer interface applications. J Neural Eng 3:139–144

Rosenstiel W, Bogdan M, Sp M (2012) Principal component based covariate shift adaption to reduce non-stationarity in a MEG-based brain–computer interface. EURASIP J Adv Signal Process 129:2–8. doi:10.1186/1687-6180-2012-129

Satti A, Guan C, Coyle D, Prasad G (2010) A covariate shift minimization method to alleviate non-stationarity effects for an adaptive brain–computer interface. In: Proceedings—international conference on pattern recognition. IEEE, pp 105–108

Shahid S, Prasad G (2011) Bispectrum-based feature extraction technique for devising a practical brain–computer interface. J Neural Eng 8:025014. doi:10.1088/1741-2560/8/2/025014

Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. J Stat Plan Inference 90:227–244. doi:10.1016/S0378-3758(00)00115-4

Sugiyama M (2012) Learning under non-stationarity?: covariate shift adaptation by importance weighting. In: Handbook of computational statistics: concepts and methods, 2nd edn. Springer, Berlin, pp 927–952

Sugiyama M, Krauledat M, Müller K-R (2007) Covariate shift adaptation by importance weighted cross validation. J Mach Learn Res 8:985–1005

Suk HI, Lee SW (2013) A novel bayesian framework for discriminative feature extraction in brain–computer interfaces. IEEE Trans Pattern Anal Mach Intell 35:286–299. doi:10.1109/TPAMI.2012.69

Tangermann M, Müller KR, Aertsen A et al (2012) Review of the BCI competition IV. Front Neurosci 6:55. doi:10.3389/fnins.2012.00055

Vapnik V (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10:988–999

Vidaurre C, Schlöogl A, Cabeza R et al (2006) A fully on-line adaptive BCI. IEEE Trans Biomed Eng 53:1214–1219. doi:10.1109/TBME.2006.873542

Wolpaw JR, Birbaumer N, McFarland DJ et al (2002) Brain–computer interfaces for communication and control. Clin Neurophysiol 113:767–791

Zhu X (2008) Semi-supervised learning literature survey. Computer science technical report 1530, University of Wisconsin, Madison

Zhu X, Ghahramani Z (2002) Learning from labeled and unlabeled data with label propagation. Technical report CMU-CALD-02-107, Carnegie Mellon University