Learning with Covariate Shift-Detection and Adaptation in Non-Stationary Environments: Application to Brain-Computer Interface

Haider Raza*, Hubert Cecotti*, Yuhua Li†, and Girijesh Prasad*
*Intelligent Systems Research Centre, University of Ulster, Londonderry, Northern Ireland, UK
†School of Computing, Science and Engineering, University of Salford, Manchester, UK
Raza-h@email.ulster.ac.uk, h.cecotti@ulster.ac.uk, y.li@salford.ac.uk, g.prasad@ulster.ac.uk

***Abstract*—Learning in the presence of dataset shifts in non-stationary environments is a major challenge. Dataset shifts in the form of covariate shifts commonly occur in a broad range of real-world systems such as, electroencephalogram (EEG) based brain-computer interfaces (BCIs). Under covariate shifts, the properties of the input data distribution may shift over time from training to test/operating phase. In such systems, there is a need for continuous monitoring of the process behavior and tracking the state of the 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. An exponential weighted moving average (EWMA) model based test is used for the covariate shift-detection in the features of EEG signals. The proposed algorithm initiates the adaptation by reconfiguring the knowledge-base of the classifier. Its performance is evaluated through experiments using a real-world dataset i.e. BCI Competition IV dataset 2A. Results show that the proposed methodology effectively performs covariate-shift-detection and adaptation and it can help to realize adaptive BCI systems.**

***Keywords*— *Non-stationary learning, dataset shift-detection, EWMA, covariate shift, adaptive learning.* `**

I. Introduction

In the machine learning literature, data are normally assumed to be drawn from stationary distribution [1]–[3]. In non-stationary environments (NSEs), the data distribution shifts over time; in general this may be due to causes such as thermal drift, ageing effects, etc. and noise. Although, non-stationary learning (NSL) algorithms have started to appear in the literature, most of them make obstructive assumptions such as high or low drift, availability of old data, and non-cyclic environments [4], [5]. 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, stock market, and wireless sensor networks. However, the aim of the NSL is to learn the evolving data that come from real-world on-line applications, and adapt to non-stationarity. When the new data is available, learn the novel part in it, and reinforce the standing knowledge that is still relevant, and overlook the past that may no longer be related, and only to be able to evoke, if and when such information becomes important again in future.

The solutions to NSL lie in devising a suitable adaptive mechanism for non-stationary systems. For such adaptive mechanisms, a few key points are given as follows: (1) the labeled data must be intelligently warehoused for classifier parameter tuning and future use, if applicable, (2) the data from the current environment is a representation of new knowledge, so it may be useful for adaptation, (3) shift-detection mechanism is required to monitor the process stationarity, and (4) the irrelevant data are required to be pruned, in such a way that significant information is not lost [2].

A Brain-Computer Interface (BCI) is an alternative communications 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 [6]–[9]. With an electroencephalography (EEG) based brain-computer interface (BCI) that operates online in real-time non-stationary/changing environments, it is required to consider input features that are invariant to shifts in the data distribution, or learning approaches that can be able to track the shifts that may repeat overtime, to update the classifier unsupervised in a timely fashion. It may be difficult to classify the EEG patterns in BCI using a traditional inductive classification algorithm, because of the non-stationarity property of the brain response characteristics in the EEG signal. The non-stationarities in the EEG maybe caused by various reasons such as changing user attention level, electrode placement, and user fatigue [1], [3]. There are notable covariate shifts in the EEG signals during trial-to-trial, and session-to-session transfers [1], [3], [10]. The covariate shift is the change in the input data distribution between training and test distribution, while the conditional distribution remains the same [11], [12].

To date, the low accuracy of classification has been one of the main concerns of the developed BCI systems based on the motor imagery detection, which directly affects the decision made by the BCI output [3]. The traditional classification algorithms are mainly inductive. To enhance the performance of the BCI system, several feature extraction, feature selection, and feature classification techniques are proposed in the literature [13]–[16]. A large variety of features have been used in BCI such as band powers, power spectral density, time frequency features, and common special patterns (CSP) based

features. However, the spatial distribution of the brain responses may change over time, resulting in shifts in feature distributions.

Various adaptive learning algorithms are present in the literature. Several studies have been conducted on adaptive BCI systems with positive results [15], [17], [18]. Most of which have made efforts to reduce the non-stationarity in the extracted features. In an adaptive learning technique, *a-priori* information is required about the shift in the EEG signal. Additionally, the adaptive techniques are mostly based on supervised learning techniques, which need labeled data, i.e. a calibration dataset.

The main drawback of the adaptation solutions proposed in the related literature is the requirement of labeled data in the operating phase. Additionally, most of the aforementioned methods are based on the batch processing for shift detection test, so there is a time delay in shift-detection. In this paper, we present a novel design methodology for an adaptive learning, which monitors the covariate shift in the input streaming data (i.e. EEG features) through our exponential weighted moving average (EWMA) model based covariate shift-detection test [10], [11] and adapts to the shift in the non-stationary conditions. The covariate shift-detection test operates in two stages, the first stage is for the shift-detection, and second stage is for shift-validation. This two-stage structure helps in reducing the false-alarms of covariate shift occurrences, which may reduce an unnecessary retraining of the classifier. An adaptation is only initiated once the shift is confirmed using validation stage, and in the adaptation stage, the classifier is retrained based on the updated knowledge base ($KB_{Upadated}$) discussed later in Section III. The proposed methodology uses two different adaptation mechanisms to update the KB of the classifier on the new knowledge. In the first method, the KB is updated incrementally using the correctly predicted labels after each shift-detection. In the second method, a transductive learning approach is used to add the relevant information into the KB. Moreover, the transductive learning is only used to increase the size of KB, but the overall classification is performed using an inductive classifier. The experiments on the real-world data are used to show that the covariate shifts can be effectively accounted for using the proposed methodology. Using the data from a BCI competition-IV 2A, we demonstrate superior performance of the proposed methodology.

This paper proceeds as follows: Section II presents a problem formulation. Next, section III presents a methodology of dataset shift-detection and validation, and covariate shift-adaptation. Section IV describes dataset and feature analysis using temporal and spatial filtering. Section V presents the experimental results and discussion. Finally, Section VI gives the conclusion.

II. Problem Formulation

Let us consider a learning framework in which the input-output pairs are defined by the $X_{Tr} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of trials, and a target output variable y_i is associated with each input vector x_i. Let us consider a two-class classification problem i.e., $y \in \{\omega_1, \omega_2\}$. The probability distribution of the inputs at time i can thus be defined as,

$$P(x_i) = P(\omega_1)\, P(x_i|\omega_1) + P(\omega_2)P(x_i|\omega_2) \qquad (1)$$

where $P(\omega_1)$, $P(\omega_2)$ are the prior probabilities of getting a sample of the class $\omega_1$and the class 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 unlabeled data from $X_{Ts} = \{(\hat{y}_i|x_i)\}_{i=1}^{M}$, where M is the number of observations in the test/operating phase.

III. Methodology

The proposed algorithm with the covariate shift–detection is a member of the family of NSL algorithms. The algorithm belongs to the category of active learning [19], where the learning model is updated at each covariate shift-detection (CSD). The CSD is performed using the CSD-EWMA test [1], [10], [11]. Its advantage is the enhanced accuracy in terms of low false-positives and low false-negatives.

A. The Algorithm Overview

The proposed algorithm is a single classifier based NSL algorithm that uses the CSD-EWMA test for initiating adaptive corrective action. It employs an active shift-detection test. The algorithm is provided with a time-series training dataset X_{Tr} ($KB_0 = X_{Tr}$) and a classifier $\mathcal{F}$ is trained. In the evaluation phase, the CSD-EWMA test is used to monitor the covariate shift and the classifier $\mathcal{F}$ is then used to classify the upcoming input data X_{Ts}.

The key elements of the proposed solution are:

- CSD_X: It monitors, the stationarity of x_i, disregarding their supervised labels.
- $\mathcal{F}$: The pattern classifier $\mathcal{F}$ is used to classify the input samples.
- $KB_{Updated}$: Updated knowledge base ($KB_{Updated}$) based on covariate shift-detection.

The proposed solution is described in Algorithm 1. After a preliminary configuration phase of the base classifier $\mathcal{F}$ and CSD_X on an initial knowledge base KB_0, the CSD_X is used to assess the process stationarity. As soon as the CSD_X detects a shift in the upcoming unlabeled data, the current learned model becomes obsolete and has to be replaced with a newly configured/retrained model. Every time, a shift is detected the new information becomes available. Based on the new information, the KB_{New} is merged with existing KB_0, and an updated KB ($KB_{Updated}$) is prepared. To prepare the $KB_{Updated}$, two methods are identified based on computational intelligence techniques: first is an adaptive learning with CSD test, and second is a transductive learning with CSD test.

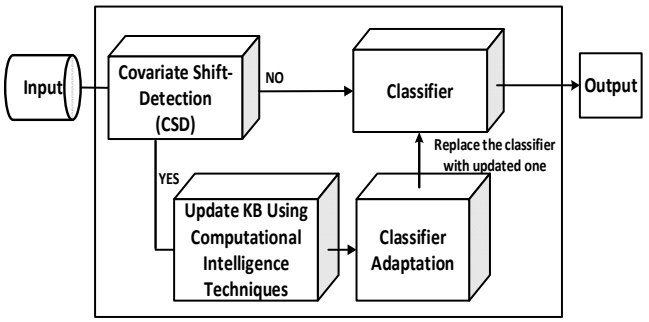


Fig 1. Architecture of adaptive learning design methodology.

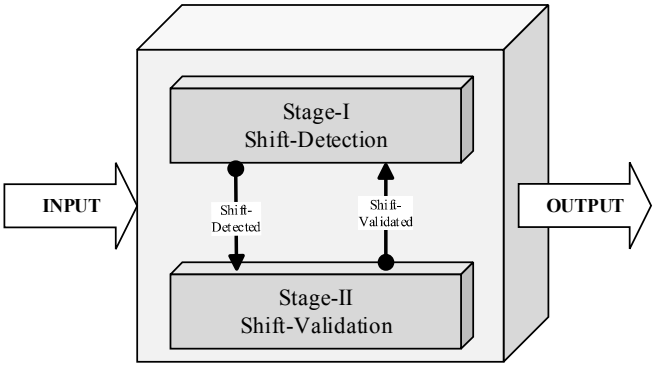


Fig 2. A two-stage covariate shift-detection (CSD). Stage-I is for shift-detection and stage-II works for validation.

The interaction between the shift-detection, validation, and classifier adaptation stages is more clearly illustrated by the Fig.1 and Fig.2, which are explained in the following subsections.

B. Shift-Detection

The first step requires a CSD test 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 to hold the supervised information in a temporary knowledge-base (KB). Since this test has to be executed online, its computational complexity maybe a critical issue. 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-EWMA [10] based test. If the test outcome at the first-stage is positive, then the second stage test gets activated, and a validation is performed in order to reduce the number of false-alarms [11]. The second stage test/validation procedure is discussed in next sub-section. The choice of the smoothing constant λ is an important issue in the EWMA based shift-detection test. In the experiments, λ is selected based on minimizing the sum of square of 1-step-ahead prediction error method.

In an EEG-based BCI, the data are generated from multiple electrodes, and hence data are multivariate. Monitoring of such processes independently maybe 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 such variables exceed their control limits simultaneously is $(0.0027)^d$, which is considerably smaller than 0.0027. So, the use of d-independent charts may provide highly distorted outcomes. A principal component analysis (PCA) is therefore used to reduce the dimensionality of the data. It provides a component, containing most of the variability in the data. This single component is used to monitor the shift in the process using SD-EWMA test [10] at the first stage .

C. Shift-Validation

According to the Algorithm 1, the KB of the classifier has to be updated at each non-stationarity shift detection. However, false positives (i.e., detection that does not correspond to an actual shift in the distribution of X) result in an unnecessary retraining. To counter this, we have introduced a shift-validation procedure as part of a two-stage structure test. This strategy aims at guaranteeing that the classifier relies on an up-to-date KB and retraining the classifier on the occurrence of a valid shift.

Algorithm 1: ALCSD

1. Configure the classifier $\mathcal{F}$ based on the initial knowledge base KB_0;
2. Configure the CSD_X using the initial knowledge base KB_0 ;
3. **FOR** $i = 1$ to length(X_{Ts})
4. Receive new data x_i;
5. **IF** (CSD_X detects a non-stationarity at time i), **THEN**
6. Update the knowledge base (KB) for classifier $\mathcal{F}$ to $KB_{Updated;}$
7. Retrain and adapt to classifier $\mathcal{F}$ on $KB_{Updated}$
8. **END**
9. Classify the input x_i by classifier $\mathcal{F}$ and get the predicted label $\hat{y}_i$ **;**
10. **END**

The shift-validation procedure exploits two sets of observations generated before and at the covariate shift-detection time point. The observations from the X_{Tr} (KB_0) are assumed to be in its stationary state, and are compared with data from the current trial in which lies the covariate shift-detection time point. To validate the shift-detection from the stage-I, the multivariate Hotelling's T-Square statistical hypothesis test is used. If the p-value of the test is below 0.05, then the shift is confirmed, otherwise it is considered as a false-alarm. On each shift-detection, the KB_{New} gets updated based on the current shift in the data.

D. Covariate Shift-Adaptation

Once the shift-detection is validated, the adaptation phase starts (see Fig. 1). To adapt to the shift, re-training of the classifier is required on the $KB_{Updated}$. In order to retrain the classifier, an additional set of input target pairs is necessary to prepare the $KB_{Updated}$. To get the set of input target pairs, we have investigated two ways of incrementally predicting target labels. In the first scenario, the labels after each trial is stored, this forms a temporary KB. Next, the predicted labels are also available. To select the meaningful information, only correctly predicted labels are added into KB_{New}. Once, the shift is detected, the classifier is re-trained on this $KB_{Updated}$ and this updated classifier is used for further classification. This

approach is quite similar to co-training [20] used in a semi-supervised learning (SSL), where the predicted labels are used to train the other classifier. In the second case, we have applied a transductive-inductive learning model to adapt to covariate shift. However, transduction is only used to add new trials into the knowledge base, and an inductive classifier is used to classify the upcoming samples. The transduction will only start once the covariate shift is detected and validated.

Both the methods mentioned above used to adapt to the covariate shift are presented below.

E. Adaptive Learning with CSD (ALCSD)

In ALCSD, initially an inductive classifier $\mathcal{F}$ is trained on the KB_0. The KB_0 consists of an N-number of labeled trials. Using KB_0, the parameter λ is obtained for the shift-detection test. Then, an evaluation phase starts, and unlabeled features are processed sequentially for classification from X_{Ts}. The shift-detection test is used to monitor the covariate shift. Once the shift is detected, it acts as an alarm to update the classifier. To update the classifier, new knowledge is required. In order to obtain a new KB, it is assumed that after each trial, the true labels are available, and among all predicted labels only correctly predicted labels are added into KB_{New}. The KB_0 and KB_{New} are merged to form a $KB_{Updated}$. The $KB_{Updated}$ is used to retrain the classifier, and further this updated classifier is used to classify the upcoming data. On each shift-detection, KB gets updated and a classifier is built and adapted incrementally.

F. Transductive Learning with CSD (TLCSD)

TLCSD model is based on a probabilistic K-nearest neighbor's (KNN's) principle. Initially, an inductive classifier $\mathcal{F}$ is trained on the KB_0 and the parameter λ for the shift-detection test is obtained. Once the classifier $\mathcal{F}$ is trained and optimal classification decision boundary is obtained, then an evaluation phase starts. First the parameters CR_{Thres}, λ and K are set, wherein CR_{Thres} is a confidence ratio threshold which will be explained later in this section, and K is the number of neighbors for transductive learning. In the evaluation phase, the classifier classifies the features obtained from the testing data $X_{Ts} = \{\hat{y}_i \mid \mathcal{F}(x_i)\}_{i=1}^{M}$. The classifier initiates adaptation through transduction after every shift-detection. Each time the classifier initiates adaptation, 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 initially from the start of evaluation phase to first detection point. Once the adaptation is initiated for 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:

$$d_{(p,q)} = \|x_p - x_q\| \tag{2}$$

where m is the number of features. This provides a vector variable $\mathbf{D} = [d_{(p,q_1)}, \dots, d_{(p,q_N)}]$, which is a vector of Euclidean distances from unlabeled data point to the N number of labeled data points. Then, the K nearest neighbors are selected (for example, if number of neighbors K=6, and $\boldsymbol{D_K} = [d_{(p,q_1)}, \dots, d_{(p,q_K)}]$ is a vector of Euclidean distances sorted in the ascending order, and $\boldsymbol{L_K} = [l_{(q_1)}, \dots, l_{(q_K)}]$ are the corresponding labels $l_{1:K} \in \{\omega_1, \omega_2\}$). For each of the K nearest points, an RBF kernel is used to compute the weight, as given in equation (3).

$$K(p,q) = exp\left(-\frac{\|x_p - x_q\|^2}{2\sigma^2}\right) \tag{3}$$

From equation (3), we have $0 \le K(p,q) \le 1$. A higher value of weight implies the data-point's closeness to the unlabeled current feature. The vector $\boldsymbol{D_K}$ contains the Euclidean distances of K nearest points and then the weights for each neighbor is given by,

$$R(i) = K(p, q_i) \tag{4}$$

Moreover, in the initial KB_0, the labels (l) are known. Using $R(i)$, for each of the classes a confidence ratios CR_{ω_i} is obtained by,

$$CR_{\omega_1} = P(\omega_1|x) = \frac{\sum_{i=1}^{K} R(i) * (l(i) == \omega_1)}{\sum_{i=1}^{K} R(i)} \tag{5.a}$$

$$CR_{\omega_2} = P(\omega_2|x) = \frac{\sum_{i=1}^{K} R(i) * (l(i) == \omega_2)}{\sum_{i=1}^{K} R(i)} \tag{5.b}$$

The confidence ratio CR_{ω_i} attained from equation (5.a & 5.b) is a posterior probability of the class membership of the current unlabeled data point. This CR_{ω_i} acts as a belief or confidence, for a data sample to belong to a particular class. In this step, for each observation from Δm, CR_{ω_i} is obtained and is used to decide if the trial's features and the estimated output labels should be added to the existing knowledge-base i.e. if $\max(CR_{\omega_1}, CR_{\omega_2}) > CR_{Thres}$ as $CR_{\omega_1} + CR_{\omega_2} = 1$, then keep the example into KB_{New}, otherwise discard it. The labels $\hat{y} = \mathcal{T}(\Delta m)$ obtained through transductive inference $\mathcal{T}$, which are above the CR_{Thres} are thus inserted into the KB_{New}. This KB_{New} is then merged into existing KB_0 (i.e. labeled data (X_{Tr})). Based on the updated KB, the inductive classifier function is updated, and a new classifier $\mathcal{F}$ is obtained. Every time KB_{New} is available, the classifier $\mathcal{F}$ is updated, and this process repeats until all the M points in the testing phase are classified.

Comparative evaluations of these methods are given in results and discussion sections.

IV. Datasets and Feature Analysis

A. Data Description: BCI Competition IV dataset 2A

The BCI Competition IV dataset 2A [21] is comprised of EEG data collected from nine subjects, namely [A01-A09], that were recorded during two sessions on separate days for each subject. The data consists of 25 channels, include 22 EEG channels, and 3 monopolar EOG channels. Among the

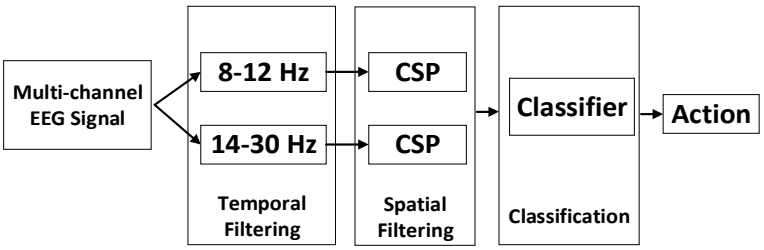
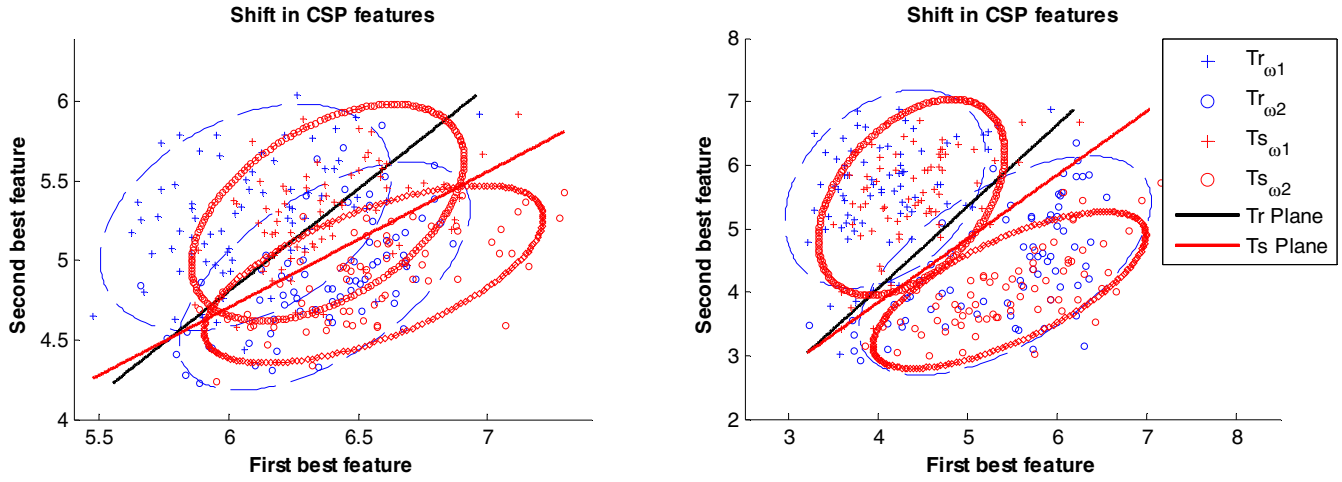


Fig.3: Block diagram for the MI based BCI. It consist of following four stages: Initially the multichannel EEG signals are acquired, next the band-pass filtering is performed, and then the CSP features are obtained to be classified using a pattern classifier. Finally, the action is performed.



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

Fig.4: 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 circle 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.

22 EEG channels, 10 channels are selected for this study, which are responsible for capturing most of the motor imagery (MI) activities. The data was collected on four different MI 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, each run comprised of 48 trials (12 for each class). The total number 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. The MI data from the session-I was used to train the classifiers, and the MI data from the session-II was used as the test. Each trial is a complete paradigm of 7.5 seconds.

B. Data Processing and Feature Extraction

a) Temporal Filtering

The second stage of the MI based BCI block diagram (see Fig.3) employs two filters that decomposes the EEG signals into two different frequency bands. The band-pass filters are used, namely [8-12] Hz (μ band), [14-30] Hz (β band). These frequency ranges are used, because these cover a stable frequency response over the range of 8-30 Hz. In the next sections, we consider a time segment of 3 s after the cue onsets for both data sets.

b) Spatial Filtering

The third stage of the MI based BCI block diagram (see Fig.3) 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 [24]. The neurophysiological origin of sensorimotor 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) [24], [25] .

A pair of band-pass and spatial filters in the second and third stages perform spatial filtering of the 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 data-driven supervised decomposition of signals parameterized by a projection matrix $W \in \mathbb{R}^{C\times C}$, where C is the number of selected channels. W projects the single trial EEG signal $E \in \mathbb{R}^{C\times T}$ in the original sensor space to $Z \in \mathbb{R}^{C\times T}$, which lives in the surrogate sensor space, as follows:

$$Z = WE \tag{6}$$

where E is a C×T EEG measurement data of a single trial, and T is the number of time points per channel. The rows of W are the spatial filters and the columns of W^{-1} are the common spatial patterns. The spatially filtered signal Z given in eq. (6) maximizes the difference in the variance of the two classes. A CSP analysis is applied in order to obtain an effective discrimination of mental states that are characterized by ERD/ERS effects. However, the variances corresponding to only a small number of spatial filters are generally used. The m first and m last rows of Z i.e. Z_t, t∈ $\{1 \dots 2m\}$ from the feature vector x_t given in eq. (11) as input to a classifier where m=1. The CSP features of the single trial are then given by:

$$x_t = log\left(\frac{var(Z_t)}{\sum_{i=1}^{m} var(Z_t(:,i)) + var(Z_t(:,C+1-i))}\right) \tag{7}$$

Then, the CSP based features from two frequency bands are combined to form the input features for a single classifier. Fig.4 shows the features obtained by a CSP technique for the subject A03. Each of sub-figures Fig. 4(a) and Fig. 4(b), represents a set of CSP features corresponding to one of the frequency bands μ and β. The blue crosses and blue circles denote the features of the left hand and right hand MI for the training data, and the red crosses and red circles denote the features of the left hand and right hand MI for the testing data, respectively. The black and red lines represent the separation planes between the features of two classes from training and test data, respectively obtained by a linear discriminant analysis (LDA) classifier. These separation planes and ellipses are plotted for the illustration purposes only, to show the covariate shift in the CSP features.

V. Experiments

In order to evaluate the performance of the system, we have considered the classification accuracy as the measure of index. The experiments are performed using a support vector machines (SVMs) pattern classifier $\mathcal{F}$. The classification accuracy is given in percent (%). The parameter λ, K and CR_{Thres} are required to be carefully selected, where λ is a smoothing constant for the shift-detection test, CR_{Thres} is a confidence ratio threshold, and K is the number of neighbors for the transductive learning. In the dataset 2A, the session-I is divided into two parts, the first 80% is used for training the pattern classifier, and second 20 % is used to obtain the optimized parameters. The evaluation is then performed on the data from the session-II. The results for the shift-detection and validation are obtained using CSD-EWMA test [1], [10], [11]. In first stage, the SD-EWMA test is used, and at the second stage, a multivariate Hotelling T-square is used for validation. The results from the shift-detection and validation is presented in the Table.1.

TABLE I
Results for Shift-Detection & Validation

Subject	Lambda	Shift-Detected	Shift-Validated
A01	0.90	5	1
A02	0.80	6	4
A03	0.10	3	1
A04	0.90	4	2
A05	0.90	4	2
A06	0.10	4	1
A07	0.90	3	2
A08	0.10	4	1
A09	0.50	3	1

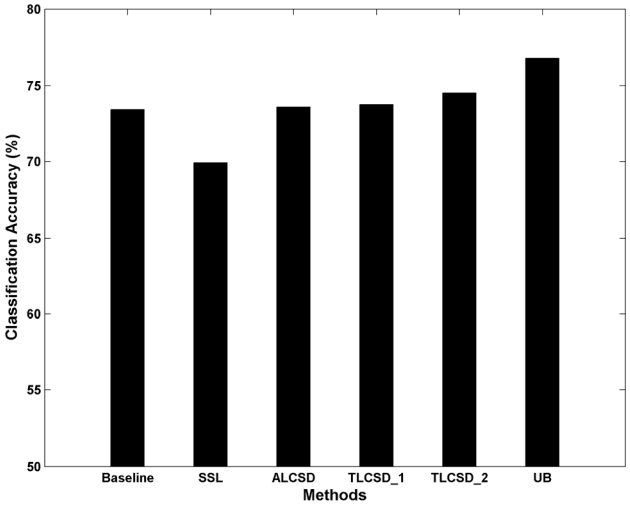


Fig.4: Results for the comparison of the mean accuracy for the proposed methods against the baseline, SSL, and optimal methods.

For each subject, a 10-fold cross-validation training accuracy is computed. The baseline method is a traditional inductive learning with CSP [26]. It does not adapt/re-train its pattern classifier. It only obtains its global classification function once during training, and remains fixed during the evaluation phase. Moreover, to compare with other methods, an semi-supervised learning (SSL) label propagation algorithm [27] have been considered and implemented. The idea behind the SSL label propagation is based on a graph to spread the labels from labeled examples to the whole unlabeled data [27]. The SSL is considered because, we have used the smoothness assumption to implement the transductive algorithm (i.e., the points which are closest to each other are more likely to share the same labels). Moreover, the two variants for the TLCSD are presented, where for the $Trans_1$, a parameter CR_{Thres} is fixed to 0.70, and for $Trans_2$, the parameter K and CR_{Thres} are subject specific, and selected based on an empirical study. The last column of the Table.2 provides an upper bound (UB) i.e. maximum classification evaluation accuracy obtained by training the pattern classifier on both the train and the test data, and evaluation is performed on the test data.

A. Results

In Table I, the results for the choice of λ and the shift-detection are presented. The choice of the smoothing constant

TABLE I
CLASSIFICATION ACCURACY (%) RESULTS FROM BCI COMPETITION IV-DATASET 2A

	10-Fold-Cross-Validation	Baseline	SSL Label	ALCSD	$TLCSD_1$		$TLCSD_2$		Upper bound
					K=18 CR_{Thres}=0.70	K	CR_{Thres}		
Subject	**Training**	**Eval**	**Eval**	**Eval**	**Eval**			**Eval**	**Eval**
A01	85.71	89.58	79.17	89.58	89.58	6	0.70	89.58	90.28
A02	75.71	53.47	54.17	54.86	56.25	18	0.60	57.64	58.33
A03	92.86	92.36	93.06	93.75	92.36	6	0.50	95.14	97.22
A04	77.86	64.58	68.06	65.97	65.28	18	0.60	65.97	67.36
A05	61.43	59.03	45.14	57.64	59.72	6	0.70	59.72	59.03
A06	71.43	65.28	56.94	64.58	65.28	18	0.70	65.28	65.97
A07	84.29	59.72	54.17	59.72	59.72	18	0.55	59.72	70.83
A08	93.57	91.67	90.97	90.97	90.28	6	0.85	91.67	91.67
A09	80.00	85.42	87.50	85.42	85.42	18	0.65	86.11	90.28
Mean	**80.32**	**73.46**	**69.91**	**73.61**	**73.77**			**74.54**	**76.78**
Std	**10.25**	**15.94**	**18.22**	**15.97**	**15.21**			**15.66**	**13.11**

λ is obtained by minimizing the sum of squares of 1-step-ahead prediction errors. The subject A02 has the maximum number of shift-detections (i.e., 6), and subjects A03, A07, and A09 have minimum number of shift-detections (i.e. 3). After the shift-validation stage, for the subject A06, the number of shift-detection has decreased from 6 to 4. This validation stage thus helps to decrease the rate of false-positive at stage-II, consequently the effort of unnecessary retraining the classifier is also reduced.

In Table-II, the 10-fold cross-validation accuracy presents the training accuracy with a mean of 80.32%, while the subject A08 has a maximum accuracy of 93.57% and the subject A06 has the worst accuracy of 71.43%. For the baseline results, only an inductive classifier is used on the test data without any adaptation on the CSP features. The baseline method gives 73.46 % of mean accuracy and subject A03 has the best accuracy 92.36%. The SSL based label propagation method gives 69.91% mean accuracy, which is below the accuracy of the baseline method. In ALCSD method, the results have shown a slight improvement in the performance against the baseline method with the mean accuracy of 73.61%, only subjects A02, A03, A04 have shown improvements. Next, for $TLCSD_1$, the parameters K and CR_{Thres} have been fixed to K=18 and CR_{Thres}=0.70, and the classification accuracy has improved slightly from 73.46% to 73.77%. Next, for $TLCSD_2$, the subject specific parameters are selected and the accuracy has improved from 73.46% to 74.54%. The subjects A02, A03, A04, A08, and A09 have shown improvements. In the last column of the Table-II, the maximum classification accuracy is obtained by training the pattern classifier on both the train and the test data, and evaluated on test data. Thus the mean classification accuracy of 76.78%, is an upper bound. Fig.5 shows the results for the classification accuracy comparison using bars representation.

B. Discussion

In the NSL, balancing the trade-off of covariate shift and adaptation is a challenging issue. Due to the covariate shift, low classification accuracy is one of the main concerns of developing a practical BCI that can be placed in daily use without a constant professional support. We have tried to address this issue through new adaptive methods involving covariate shift detection and classifier adaptation.

The combination of the EWMA based covariate shift-detection and unsupervised classifier adaptation is a promising method for learning in the non-stationary environments, particularly because the shift-detection test only uses the unlabeled data for monitoring the covariate shift. Ensuring robustness of the covariate shift-detection test by appropriate selection of λ, plays an important role in initiating an adaptive action, only when it is really needed. The shift-validation at the second stage of shift detection test is demonstrated to play a crucial role in reducing the number of unnecessary classifier retraining efforts.

The proposed methods are based on the active adaptive learning, where the adaptation is only initiated once a shift is detected and validated. Once the shift is validated, the knowledge management procedure is executed to extract the meaningful information from the data. In ALCSD, only an inductive classifier is used to act as a global function and this global function is updated once the new information is available. Whereas, in TLCSD, two learning methods are considered, first is an inductive learning that is responsible for overall classification, and the second is a transductive learning that is only used to add more trials (or input-target pairs) into the KB. This transductive approach helps to track the evolution of the shift and adds new information when there is a significant shift in the distribution of the data at operational stage. In TLCSD, the parameters K and CR_{Thres} are required to be carefully tuned. In the $TLCSD_1$, the parameters are selected based on an empirical study and it shows that the predicted label through transductive learning for which the

confidence is greater than 70% is only added to the KB. However, this choice may not be best for all the subjects, so to select the subject specific parameters, the $TLCSD_2$ is presented and it is shown to achieve a better accuracy.

The experimental results demonstrated the effectiveness of the proposed covariate shift detection and adaptation methods. The results also showed that the learning with proposed method has outperformed the traditional learning methods and SSL with CSP filters.

VI. 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 (ALCSD and TLCSD) are 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 is evaluated on a multi-variate cognitive task detection in EEG-based BCI as part of the BCI Competition IV dataset 2A and better results in terms of increased classification accuracy are obtained. The ALCSD has shown only a slight improvement, whereas the TLCSD has shown a good improvement. More detailed experimental analysis shows that the performance of the proposed method are better in a range of non-stationary situations. This work is planned to be extended further by employing the CSD into the task of fault monitoring as well.

Acknowledgement

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 University of Ulster. 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).

References

[1] H. Raza, G. Prasad, and Y. Li, "EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments," *Pattern Recognit.*, vol. 48, no. 3, pp. 659–669, Aug. 2014.

[2] H. Raza, G. Prasad, and Y. Li, "Adaptive learning with covariate shift-detection for non-stationary environments," in *IEEE14th UK Workshop on Computational Intelligence (UKCI)*, 2014, pp. 1–8.

[3] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, "Application of Covariate Shift Adaptation Techniques in Brain-Computer Interfaces.," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1318–24, Jun. 2010.

[4] R. Elwell and R. Polikar, "Incremental Learning in Nonstationary Environments with Controlled Forgetting," in *Proceeding of International Joint Conference on Neural Networks*, 2009, pp. 771–778.

[5] J. Gama and P. Kosina, "Recurrent concepts in data streams classification," *Knowl. Inf. Syst.*, May 2013.

[6] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A Review of Classification Algorithms for EEG-Based Brain-Computer Interfaces.," *J. Neural Eng.*, vol. 4, no. 2, pp. R1–R13, Jun. 2007.

[7] M. Thulasidas, C. Guan, S. Member, J. Wu, and A. P. Speller, "Robust Classification of EEG Signal for Brain Computer Interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 1, pp. 24–29, 2006.

[8] K. Müller, M. Krauledat, and G. Dornhege, "Machine learning techniques for brain-computer interfaces," *J. Biomed. Eng.*, vol. 49, pp. 11–22, 2004.

[9] H. Raza, G. Prasad, Y. Li, and H. Cecotti, "Toward Transductive Learning Classifiers for Non-Stationary EEG," in *Engineering in Medicine and Biology Society (EMBC), 2014 35th Annual International Conference of the IEEE*, 2014.

[10] H. Raza, G. Prasad, and Y. Li, "Dataset Shift Detection in Non-stationary Environments Using EWMA Charts," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 3151–3156.

[11] H. Raza, G. Prasad, and Y. Li, "EWMA Based Two-Stage Dataset Shift-Detection in Non-stationary Environments," in *Artificial Intelligence Applications and Innovations.*, 2013, pp. 625–635.

[12] H. Shimodaira, "Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function," *J. Stat. Plan. Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000.

[13] H. Suk and S. Lee, "A novel Bayesian framework for discriminative feature extraction in brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 286–299, 2013.

[14] L. Kuncheva and W. Faithfull, "PCA feature extraction for change detection in multidimensional unlabeled data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 69–80, 2014.

[15] C. Vidaurre, R. Cabeza, R. Scherer, and G. Pfurtscheller, "A Fully On-line Adaptive BCI," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 1214–1219, 2006.

[16] D. Coyle, G. Prasad, and T. M. McGinnity, "Faster Self-Organizing Fuzzy Neural Network Training and A Hyperparameter Analysis for A Brain-Computer Interface.," *IEEE Trans. Syst. Man, Cybern.*, vol. 39, no. 6, pp. 1458–71, Dec. 2009.

[17] J. Blumberg and J. Rickert, "Adaptive classification for brain computer interfaces," in *29th Annual International Conference of the IEEEEngineering in Medicine and Biology Society, EMBS*, 2007, vol. 1, no. 4, pp. 2536–2539.

[18] P. Shenoy, M. Krauledat, B. Blankertz, R. P. N. Rao, and K.-R. Müller, "Towards Adaptive Classification for BCI.," *J. Neural Eng.*, vol. 3, no. 1, pp. R13–23, Mar. 2006.

[19] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Non-Stationary Environments.," *IEEE Trans. Neural Networks*, vol. 22, no. 10, pp. 1517–31, Oct. 2011.

[20] X. Zhu, "Semi-supervised learning literature survey," *Comput. Sci. Univ. Wisconsin-Madison*, 2006.

[21] R. Leeb, C. Brunner, Müller-Putz, A. G. R., Schlögl, and G. Pfurtscheller, "BCI Competition 2008–Graz data set B. Graz University of Technology, Austria." 2008.

[22] C. Brunner, R. Leeb, and G. Müller-Putz, "BCI Competition 2008–Graz data set A." pp. 1–6, 2008.

[23] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the BCI Competition IV," *Front. Neurosci.*, vol. 6, p. 55, Jan. 2012.

[24] B. Blankertz and R. Tomioka, "Optimizing spatial filters for robust EEG single-trial analysis," *Signal Processing Magazine, IEEE*, no. Jan 2008, pp. 41–56, 2008.

[25] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b," *Front. Neurosci.*, vol. 6, p. 39, Jan. 2012.

[26] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter Bank Common Spatial Pattern (FBCSP)," in *Proc. Int'l Joint Conf. Neural Networks*, 2008, pp. 2390–2397.

[27] X. Zhu and Z. Ghahramani, "Learning from Labeled and Unlabeled Data with Label Propagation," Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.