Adaptive Learning with Covariate Shift-Detection for Non-Stationary Environments

Haider Raza, Girijesh Prasad, Yuhua Li
Intelligent Systems Research Center, University of Ulster, Londonderry, Northern Ireland, UK
raza-h@email.ulster.ac.uk, g.prasad@ulster.ac.uk, y.li@ulster.ac.uk

***Abstract*—Learning with dataset shift is a major challenge in non-stationary environments wherein the input data distribution may shift over time. Detecting the dataset shift point in the time-series data, where the distribution of time-series shifts its properties, is of utmost interest. Dataset shift exists in a broad range of real-world systems. In such systems, there is a need for continuous monitoring of the process behavior and tracking the state of the shift so as to decide about initiating adaptation in a timely manner. This paper presents an adaptive learning algorithm with dataset shift-detection using an exponential weighted moving average (EWMA) model based test in a non-stationary environment. The proposed method initiates the adaptation by reconfiguring the knowledge-base of the classifier. This algorithm is suitable for real-time learning in non-stationary environments. Its performance is evaluated through experiments using synthetic datasets. Results show that it reacts well to different covariate shifts.**

Index terms- Non-stationary learning, dataset shif-detection, EWMA, covaraite shift, adaptive learning. `

I. INTRODUCTION

IN real-world machine learning applications, processes are often characterized by an evolving nature and may shift their behaviour over time. In general this may be due to thermal drift, ageing effects, and non-stationary environments (NSEs). These effects and faults may adversely affect environmental, natural, artificial and industrial processes. Even if no shift has occurred, the evolving process might provide additional information that could be exploited to enhance the system accuracy. In all real-world applications, non-stationarity is quite common, especially with the systems interacting with the dynamic and evolving environments, e.g., data coming from wireless sensor networks, and electroencephalogram (EEG) based brain-computer interfaces. In the stationary case, the integration of fresh information requires a supervision mechanism to improve the accuracy of a classification system. For instance, in quality analysis applications, when an expert or supervisor is present to detect the artifacts, outliers, and false-positives, such information is useful to enhance the accuracy of the system by taking an appropriate corrective action. However, an expert for labelling and monitoring the data is expensive and it requires a lot of manual efforts, which maybe too difficult to undertake specially for data-intensive real-time systems.

The solutions therefore lie in devising an appropriate adaptive mechanism for non-stationary systems. For such adaptive mechanisms, a few key points are given as follows: (1) the data samples 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) the shift-detection or process monitoring mechanism is required to check the stationarity of the process, and (4) pruning of irrelevant data is required to be done in such a way that no relevant information is lost.

Traditionally classifiers [1]–[6] are built upon the common assumption that the data distribution remains stationary over training and testing phases. Their performance is therefore adversely affected in non-stationary conditions.

There exists a large literature addressing the non-stationary learning, with research focusing on adaptive pre-processing techniques, adaptive neural networks, and adaptive classifiers for specific applications. In an adaptive learning algorithm called floating approximation in time-varying knowledge-base (FLORA) [7], an adaptive windowing based learning algorithm in the presence of concept drift is presented. Similarly, in [8] an adaptive sliding window (ADWIN) algorithm with drift detection approach is introduced. It monitors the concept shift and the online error. The drawbacks of the algorithm are excessive time and memory requirements. Later in [9] an ADWIN2 algorithm is proposed with low memory and time consumption. In [10] a just-in-time (JIT) adaptive classification based on temporal shift-detection of process deviation was proposed. This method detects the shift in the data generating process and once the shift is detected, an adaptive management of knowledge base (KB) is executed. Later, in [11] a JIT adaptive classification based on the intersection of confidence interval (ICI) rule was presented, a key good feature of this method is that no assumption is made about the distribution of data generating process. The ICI rule has better detection ability by its hierarchical structure that validates the shift, which has occurred due to the variation in the process and not because of noise. In [12], JIT based ensemble of classifiers was presented, this method assesses the stationarity in both the classification error and unlabeled data. This method handles recurrent concepts within an ensemble of classifiers framework. In [13] an incremental learning in concept shift for non-stationary environments is presented. In this a Learn++.NSE algorithm is presented which tracks the shifting environments, regardless of type of concept shift. A learning algorithm for recurrent concepts based on JIT family of classifiers is presented in [14], it also uses the shift-detection test on both classification error and unlabeled data. Recently, a semi-supervised learning framework for initially labeled non-

978-1-4799-5538-1/14/$31.00 ©2014 IEEE

stationary streaming data is presented; the method is known as compacted object sample extraction (COMPOSE) [15].

The main limitation of the solutions proposed in the related literature is the requirement of supervised data samples during operational lifetime. Furthermore, most of the previous methods are based on the batch processing for shift detection test, so there is a time delay in shift-detection. Hence, those batch processing methods are not so useful for real-time systems where initiating adaptation in the nick-of-time is of paramount interest.

Here, we propose a design methodology for an adaptive classification method which, monitors the covariate shift in the input streaming data through our exponential weighted moving average (EWMA) model based shift-detection test [16], [17] and reacts to the shifting environments in the non-stationary conditions. Based on the shift-detection point, an adaptation is initiated through retraining of the classifier based on the updated knowledge base ($KB_{Upadated}$) discussed later in Section III. The proposed method uses different adaptation mechanisms to retrain the classifier on the new and initial knowledge base. It is demonstrated to outperform a traditional learning approach without any shift-detection test. The approach is computationally efficient because of low computational cost and less memory requirements during online processing. So, this scheme can be deployed along with any base classifier such as k-nearest neighbour (kNN), support vector machine (SVM), or Naïve Bayes (NB) in an adaptive learning algorithm.

This paper proceeds as follows: Section II presents a background of dataset shift-detection and non-stationary learning. Section III is a problem formulation; Section IV consists of proposed methodology with the adaptive classification algorithm. Section V presents the datasets used in the experiment. Finally, Section VI shows the experimental results and discussion.

II. BACKGROUND

Dataset shift: The term dataset shift [18], [19] was first defined in the workshop of neural information processing systems (NIPS, 2006). The dataset shift is a "*case where the joint distribution of inputs and outputs differs between training and test stage, i.e., when* $(P_{train}(y,x) \neq P_{test}(y,x))$" [20]. Dataset shift was previously defined by various authors giving different names to the same concept such as, concept shift or drift [7], changes of classification [21], changing environment [22], contrast mining [23], and fracture point [24]. In pattern classification problems, the dataset shift is now mainly categorized into three different types that usually occur in the real-world applications such as (i) *covariate shift*, (ii) *prior probability shift,* and (iii) *concept shift*.

Covariate Shift: The covariate shift has been defined by different terms in the literature. Several authors defined covariate shift as, "population drift", "a case where the population distribution may change over time" [4]. In a generic way, it is defined as "covariate shift appears only in $X \rightarrow Y$ problems, and the case where the conditional probability in training and testing remains same $(P_{train}(y|x) = P_{test}(y|x))$, but the input distribution $P(x)$ changes between training and testing, i.e., $(P_{train}(x) \neq P_{test}(x))$" [19]. Let's take an example of a process where covariate shift can be seen. Assume a training input data distribution $P_{train}(x)$ is a normal distribution with mean and standard deviation as 2 and 1.5 respectively, i.e. $[x_{train} = \mathcal{N}(x; 2,1.5)]$ and the test input data distribution $P_{test}(x)$ is also a normal distribution with mean and

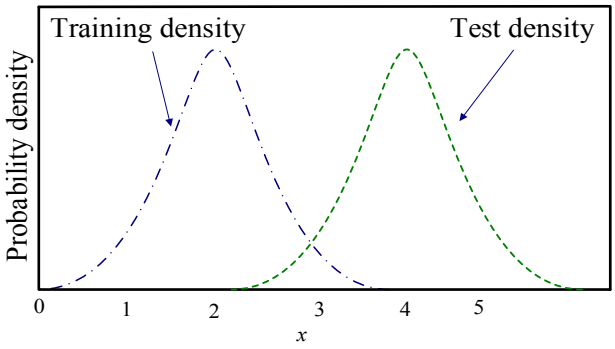


Fig. 1. *Covariate shift*: Training dataset has normal distribution with $\mathcal{N}(x; 2, 1.5)$, and test dataset also has normal distribution with $\mathcal{N}(x; 4, 1.5)$. Thus the mean of the testing data distribution has changed from that of training, resulting in covariate shift.

the standard deviation as 4 and 1.5 respectively, i.e. $[\ x_{test} = \mathcal{N}(x; 4,1.5)]$. Fig. 1 shows the covariate shift as is given in the example above where only the mean has changed between the training and test stages.

The problem of covariate shift can be easily found in the real-world applications. Some of the common examples are spam filtering, brain-computer interfaces (BCIs), and network intrusion detection. For other types of dataset shift such as *prior probability* and *concept shift*, readers may refer to [16], [17]. There exists other shifts that could happen in theory, but we are not discussing those as they appear rarely, for more details see [19]. In this paper, our main focus is on the covariate shift-detection (CSD), because the pattern classification problem is based on the predictive model, i.e., $X \rightarrow Y$.

Shift-Detection Test (SDT): To assess the stationarity of the data generating process, a shift-detection test is required. This paper uses a covariate shift-detection test based on a two-stage structure [16], [17]. The first stage works in an online mode and it uses an exponentially weighted moving average (EWMA) model based control chart to detect the covariate shift-point in non-stationary time-series. The second stage validates the shift detected by the first stage using the Kolmogorov–Smirnov (K–S test) statistical hypothesis test.

Non-stationary Learning (NSL): The algorithms designed for NSL may be categorized in several ways:

1) Online vs. batch algorithms
2) Single vs. ensemble of models based approaches
3) Active vs. passive approaches

Online learning algorithms learn one sample (instance) at a time, whereas in batch learning a chunk of instances or samples are required. Online learning has better plasticity (i.e., learning new knowledge) and poor stability (i.e., retaining existing, relevant and recurring knowledge) properties. In NSL, this trade-off of plasticity and stability need to be balanced. The online NSL algorithms are more sensitive to noise. In batch learning, the size of batch plays an important role, as large

amount of data have better stability but learning can be ineffective if the size of the batch is too small. Another possibility of poor learning is when the data are coming from the multiple environments in the same batch. In batch learning, the windowing approach may be used to control the batch size. The example of this approach is an instance selection using single classifier e.g., STAGGER [25] and FLORA [7]. These algorithms use a sliding window approach to select a batch and train a new classifier. In some approaches it is suggested to vary the size of the window based upon some heuristics such as 'how fast the environment is changing'. The FLORA algorithm has a built-in forgetting method for the information falling outside the window. More recently, there have been several modifications to this window based approach, each has its own heuristics such as combining the shift-detection test with learning, putting the choice on the classifier, or introducing the error threshold. Most recently, the approach involving ensemble of classifiers [14] combines multiple hypotheses in hope to form a better hypothesis. The main idea behind this is to combine many local learners in an attempt to produce a strong global learner. In active learning, a shift-detection mechanism is included [10], [11] and the learning model will only be updated, once a shift is detected. Whereas, in passive learning, the model updates continuously with each new dataset [13].

III. Problem Formulation

Let us consider the adaptive learning framework in which inputs x_i is the observation data generated from the process X according to unknown distribution in the time period i. A target variable y is associated with x_i. Let us consider a two-class classification problem i.e., $y \in \{y_1, y_2\}$. The probability distribution of the inputs at time i can thus be defined as,

$$P(x|i) = P(y_1|i)P(x|y_1, i) + P(y_2|i)P(x|y_2, i) \qquad (1)$$

where $P(y_1|i), P(y_2|i)$ are the prior probabilities of getting a sample of class y_1 and y_2, respectively, while $P(x|y_1, i), P(x|y_2, i)$ are the conditional probability distribution for the time period i. Both the prior probabilities of classes and the conditional probability are assumed to be unknown and may shift over time, whenever the non-stationarity occurs. The training sequence consists of the first I_0 observations that are assumed to be generated in stationary conditions i.e., joint distributions do not change within the time interval $[0, I_0]$. In this training period, the input target (label) pairs (x_i, y_i) are provided. The goal is to predict the labels of upcoming samples during the operating stage from I_0 to n, where n is the number of observations in the test/operating data.

IV. The Proposed Solution

The proposed algorithm *adaptive learning with covariate shift–detection* (ALCSD), is a member of NSL family of algorithms. The algorithm belongs to the category of active learning, where the learning model updates on each covariate shift-detection (CSD). The CSD is performed using TSSD-EWMA [17] test. Its advantage is the enhanced accuracy in terms of low false-positives, low false-negatives and short time-delay in the shift-detection process.

A. The Algorithm Overview

The ALCSD is a single classifier based NSL algorithm that uses CSD test for initiating adaptive corrective action. It employs an active shift-detection test. It can handle a variety of non-stationary environments, including gradual, cyclical and abrupt covariate shift. The algorithm is provided with a series of training datasets $I_i = \{x_i \epsilon X; y_i \epsilon Y\}, i = 1 \dots \dots \dots I_0$ and a classifier is trained. The trained classifier is then used to classify the upcoming input data.

The key elements of the proposed solution are:

- SDT_X: the SDT analyses the raw observations to monitor the stationarity of x_i, disregarding their supervised labels.
- K: The base classifier used to classify the input samples.
- $KB_{Updated}$: Updated knowledge base (KB) using the data with covariate shift.

The proposed solution is described in Algorithm 1. After a preliminary configuration phase of the base classifier K and SDT_X on an initial knowledge base KB_0, the SDT_X is used to assess the process stationarity. As soon as the SDT_X detects a shift in the upcoming unlabeled data, the current learned model becomes obsolete and is to be replaced with a newly configured/retrained model. Every time a shift is detected the new knowledge base (KB_{New}) gets updated. To do so, the knowledge base (KB) of the classifier K is re-trained/reconfigured by using the updated KB ($KB_{Updated}$) i.e., the merged combinations of KB_{New} and KB_0. To update the existing KB, several methods are identified as given in Table I. Once the classifier K is reconfigured using the $KB_{Updated}$, the upcoming inputs are classified.

The interaction between the shift-detection, validation and classifier adaptation stages is more clearly illustrated in the following subsections.

Algorithm 1: ALCSD

Configure the classifier K based on the initial knowledge base KB_0;
Configure the SDT_X using the initial knowledge base KB_0 ;
FOR $i = 1$ to the length of testing data
  Receive new data x_i;
  IF (SDT_X detects a non-stationarity at time i), **THEN**
    Update the knowledge base (KB) for classifier K to $KB_{Updated}$;
    Retrain the classifier on $KB_{Updated}$ as suggested in the Table I
  END
  Classify the input x_i by classifier K and get the predicted label $\hat{y}_i$;
END

TABLE I: Methods to update Knowledge base of Classifier

No.	$KB_{Updated}$: Method to update KB and retrain classifier
A	*Learning without CSD*
B	*Adaptive learning with CSD*
C	*Adaptive learning on KB_{New} with CSD*
D	*Adaptive learning on combined KB with CSD*
E	*Transductive learning with CSD*

B. Shift-Detection

The first step requires an online SDT 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 step is crucial 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 might be a critical issue. The first-stage of the test provides an initial estimate I_{ini} of the shift i.e., where the actual shift has occurred. The first-stage test is performed by SD-EWMA[16] based test. If the test outcome in the first-stage is positive, the second stage test gets activated and a validation is performed in order to reduce the false-alarms. The second stage test/validation procedure is discussed in next sub-section.

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 the up-to-date KB because the data obtained after the first detection and before the validation is from current environments. This new information (along with existing classifier predicted labels) are useful for retraining the classifier.

The shift-validation procedure exploits two sets of observations generated before and after the covariate shift time point. To this end, the SDT detects a non-stationary shift at time I_{ini} and later the validation procedure is completed at $\hat{I}$ and the shift is confirmed. The observations for the time period $[1, I_0]$, represents the process in its stationary state, are compared with those during the time period $[I_{ini}, \hat{I}]$ that represents the time period after the detection and before the end of the validation period. This interval $[I_{ini}, \hat{I}]$ is used for creating a new knowledge-base (KB_{New}). On each shift-detection, the KB_{New} gets updated based on the current shift in the data and KB_0 remains fixed. Fig 2 is an illustrative example of shift-validation.

D. Covariate Shift-Adaptation

Once the shift-detection is confirmed, the adaptation phase starts. To adapt to the shift, re-training/configuration of the classifier is required on the $KB_{Updated}$. In order to retrain the classifier, input target pairs are necessary. To get the input target pairs, the data after the initial shift is detected, is stored continuously until the validation point, which forms KB_{New} within the window $[I_{ini}, \hat{I}]$.These data points are classified using recent classifier to provide predicted labels. This approach is quite similar to co-training [26] used in semi-supervised learning, where the predicted labels are used to train the other classifier. So this new knowledge (KB_{New}) from the period $[I_{ini}, \hat{I}]$ and the initial KB_0 from the period $[1, I_0]$ are used in several ways to retrain the classifier. The $KB_{Updated}$ is prepared from the KB_{New} and KB_0. The methods explored for adapting the shift and retraining the classifier are given in Table 1 and discussed below.

A) Learning without CSD: This is the traditional learning without covariate shift-detection.

B) *Adaptive learning with CSD*: Adaptive learning with covariate shift-detection and retraining the classifier on initial KB_0 plus new knowledge KB_{New} obtained from the delay in shift-detection.

C) *Adaptive learning on KB_{New} with CSD*: Adaptive learning with covariate shift-detection and retraining the classifier only on the new knowledge KB_{New} obtained from the validation period of shift-detection. The KB_{New} is the data from a new data distribution. The idea behind this method is illustrated through Figure 3. It shows a covariate shift from training to testing in the input data distribution. The shift in the data is represented by the shaded region.

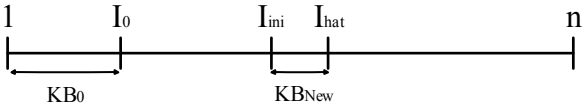


Figure 2. An illustrative example of the shift-validation for covariate shift-detection. The set of intervals, $[1, I_0]$ is initial knowledge base. The I_{ini} is the initial point where the shift is detected. $\hat{I}$ is a point where the shift has been confirmed. The interval $[I_{ini}, \hat{I}]$ is the period for new data collection.

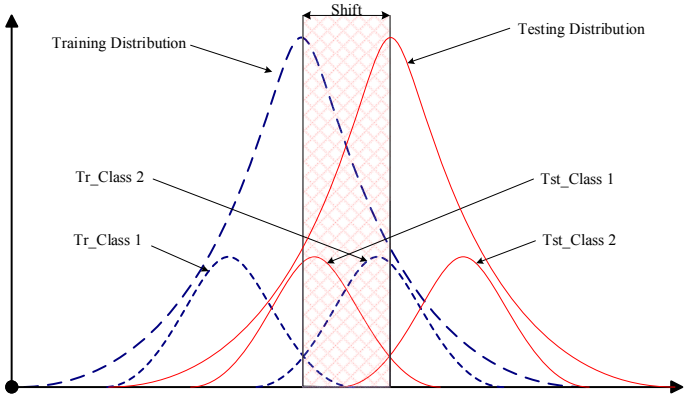


Figure 3. A covariate shift in the data. The blue dash line shows the training distribution and red dash line shows the testing distribution, where the mean is shifted. Under both distributions, there are two classes, class 1 and class 2. The shaded area is the region where the classifier has a chance for wrong prediction, after the shift in the data.

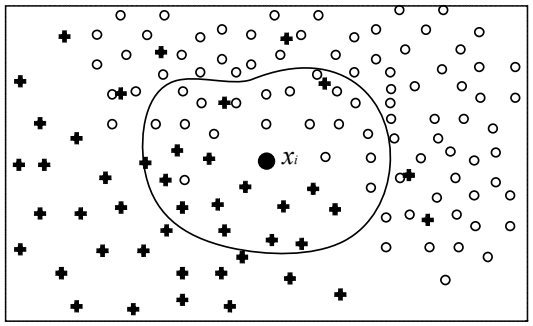


Figure 4. An illustrative example of a transductive model with sub-training dataset of neighbouring samples for each new input vector x_i. The neighbours are selected based on the Euclidean distance, where '+' and 'o' represent the class 1 and class 2, respectively.

The classification after the covariate shift is done by the classifier trained from initial KB_0, hence the chance of getting the correct label depends upon the amount of shift. The shaded area is the region where the chance of getting wrong labels are high, hence the data belonging to this area will be mostly classified wrongly. But, apart from this shaded region, still there are correctly predicted labels and retraining upon that may lead to achieve the goal of getting robust model for non-stationary learning.

D) *Adaptive learning on combined KB with CSD*: Adaptive learning with covariate shift-detection and retraining the

classifier on initial KB_0 plus the KB_{New} as given below. Receive the KB_{New} and separate the data into two classes through predicted labels. Calculate the mean of each class and compute the Euclidean distance from the mean of the classes to the corresponding classes in the initial KB_0. Sort them in ascending order and select the equal amount of data from both the KB_0 and KB_{New}.

E) *Transductive learning with CSD*: Transductive learning procedure is proposed that is based on Vapnik's principle [6] i.e., "*When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one*". So, here initial KB_0 is taken as given in method D, take the current input data x_i and compute the Euclidean distance from it to data in initial KB_0 and sort them in ascending order, then select half of the data from both the classes and merge with KB_{New} then retrain the model. This approach is explained through Fig 4., where x_i is the current input and the neighbours are selected from the initial KB_0 based upon the Euclidian distance. This approach uses the current input x_i and gathers the supervised information from the old data distribution based on the distance measure.

The comparison between these methods are given in results and discussion.

V. Datasets and Features

To assess the performance of the proposed adaptive learning algorithm, a series of experimental evaluation have been performed on synthetic dataset taken from [14]. The dataset are described as follows:

Dataset 1- One class covariate shift (D1): The dataset is consisting of 10000 data-points, the non-stationary shift occurs in the middle of the data stream at 5001 data-point, by shifting the mean of second class from $\mathcal{N}(x: 2.5, 2)$ to $\mathcal{N}(x: 4.5, 2)$, while the first class $\mathcal{N}(x: 0, 2)$ remains stationary, where $\mathcal{N}(x: \mu, \sigma)$ denotes the normal distribution with mean and standard deviation respectively.

Dataset 2-Classes swap concept shift(D2): The dataset is consisting of 10000 data points at 5001 data-point, the swap of classes occurs in the middle of the data stream, by shifting mean from $\mathcal{N}(x: 0, 2)$ to $\mathcal{N}(x: 2.5, 2)$ and $\mathcal{N}(x: 2.5, 2)$ to $\mathcal{N}(x: 0, 2)$ for class 1 and 2, respectively. This is a concept shift.

Dataset 3- Abrupt covariate shift affecting both classes (D3): The dataset is consisting of 10000 data-points, the abrupt shift affects both the classes in the middle of the stream at 5001 data-point by shifting the mean from $\mathcal{N}(x: 0,2)$ to $\mathcal{N}(x: 2, 2)$ and $\mathcal{N}(x: 2,2)$ to $\mathcal{N}(x: 4, 2)$ for class 1 and 2, respectively.

Dataset 4- Transient covariate shift (D4): The dataset is consisting of 10000 data-points, the abrupt shift effect both the classes in alternating sequence after every 2000 data-points. The shift in the mean is from $\mathcal{N}(x: 0,2)$ to $\mathcal{N}(x: 2, 2)$ and $\mathcal{N}(x: 2,2)$ to $\mathcal{N}(x: 4, 2)$ for class 1 and 2, respectively.

Dataset 5- Altering concepts shift and classes Swap (D5): The dataset is consisting of 10000 data-points, the abrupt shift and swap of classes affect the alternating sequence after every 2000 data-points. The shift in the mean is from $\mathcal{N}(x: 0,2)$ to $\mathcal{N}(x: 2, 2)$ and $\mathcal{N}(x: 2,2)$ to $\mathcal{N}(x: 4, 2)$ for classes 1 and 2, respectively and simultaneously the swap of classes occurs.

Dataset 6- Stairs sequence of covariate shifts (D6): The dataset is consisting of 10000 data-points, the stairs of shift occurs after every 2000 data-points. The shift in the mean is from $\mathcal{N}(x: 0,2)$ to $\mathcal{N}(x: 2, 2)$ and from $\mathcal{N}(x: 2, 2)$ to $\mathcal{N}(x: 4, 2)$ and so on for both the class 1 and the class 2, respectively.

VI. Experiments

The classification error i.e., the percentage of misclassification at each time instant, has been considered as an index to measure the performance of the system. Here, three base classifiers are used including k-Nearest Neighbour (k-NN), Support Vector Machine (SVM) and Naïve Bayes (NB). The average classification errors over the entire datasets are reported in Tables II, III, and IV. In each table, the classification error is computed for classifiers trained based on the predicted labels and on the supervised (actual) labels, denoted as *Pred* and *Sup*, respectively. The comparison between *Pred* and *Sup* is done to evaluate the performance of the proposed methods over the learning on the supervised labels.

A. Results

Application D1 contains an abrupt covariate shift in one class. The method C, which is learning with covariate shift-detection and retraining the classifier on KB_{New} gives the lowest classification error with all the three base classifiers.

Application D2 is a class swap occurring in the middle of the dataset with concept shift. The performance remains the same of all the methods because dataset is a concept shift.

Application D3 is an abrupt covariate shift in the middle of the data affecting both the classes. For this dataset, the method C dominates other methods with all the three base classifiers. The method C reports the lowest classification error, which is close to the transductive approach E. It is clear from Fig 5(a) that once the covariate shift occurs the classification error increases with method A, which is a traditional method. While, in Fig 5(b), the classification error decreases after the covariate-shift because of the adaptation initiated according to method C.

Application D4 has a transient covariate shifts in the data. For this dataset, the method C performs well with k-NN and SVM classifiers. The method E suits well with NB, which is a transductive approach.

Application D5 has altering concepts and classes swap with shift affecting the data. The performance remains the same of all the methods because dataset is a concept shift and there is no covariate shift-detection during the testing phase.

Application D6 is a stair of covariate shift in the mean for each step. In this case the method C performs better among all other methods. The performance of SVM classifier is the best with lowest classification error.

B. Discussion

In non-stationary learning, balancing the trade-off of plasticity and stability is a big challenge. We have tried to address this issue through few new adaptation methods in an active learning scenario. The performances of all the suggested methods are discussed below.

TABLE II: CLASSIFICATION ERROR (%) FOR DIFFERENT DATASETS USING K-NN CLASSIFIER

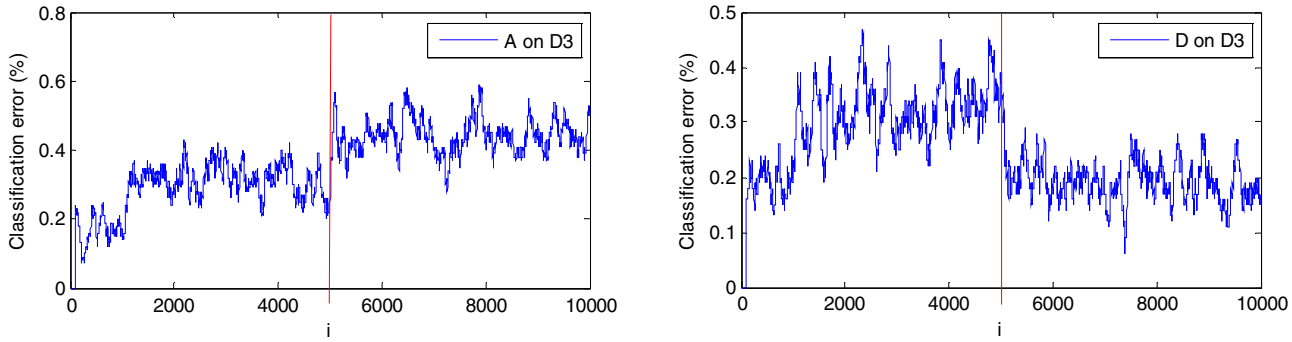
	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2414	0.2417	0.2385	0.2316	0.1989	0.2404	0.2345	0.2476	0.2373
D2	0.4806	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807	0.4807
D3	0.3301	0.3312	0.3293	0.3243	0.2987	0.3312	0.3367	0.3359	0.3343
D4	0.3190	0.3176	0.3173	0.3151	0.3322	0.3364	0.3309	0.3104	0.3295
D5	0.4492	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496	0.4496
D6	0.4290	0.4290	0.3789	0.4139	0.3154	0.4209	0.3745	0.4227	0.3726

TABLE II: CLASSIFICATION ERROR (%) FOR DIFFERENT DATASETS USING SVM CLASSIFIER

	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2452	0.2468	0.2584	0.2352	0.2328	0.2419	0.238	0.234	0.232
D2	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946
D3	0.3114	0.3151	0.3201	0.3218	0.3125	0.3133	0.3164	0.3133	0.3205
D4	0.3002	0.2891	0.3044	0.2841	0.2821	0.2889	0.2764	0.2861	0.3043
D5	0.4618	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621	0.4621
D6	0.4359	0.4158	0.4091	0.2887	0.2798	0.4199	0.3338	0.411	0.3321

TABLE II: CLASSIFICATION ERROR (%) FOR DIFFERENT DATASETS USING NB CLASSIFIER

	A	B		C		D		E	
		Pred	Sup	Pred	Sup	Pred	Sup	Pred	Sup
D1	0.2157	0.2107	0.2068	0.2002	0.1782	0.1956	0.1787	0.1962	0.1789
D2	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946	0.4946
D3	0.3103	0.3099	0.3005	0.3021	0.2687	0.3161	0.2698	0.3144	0.2697
D4	0.3004	0.3005	0.296	0.3086	0.2835	0.3009	0.2769	0.3017	0.277
D5	0.4539	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543	0.4543
D6	0.4356	0.4348	0.4318	0.4328	0.28	0.4378	0.3475	0.4363	0.3473



(a) (b)

Figure 5. The classification error as function of time for the propose method vs. traditional method. The average classification error is computed on a window containing the 100 supervised samples. (a) Classification error with method A on dataset D3. (b) Classification error with method D on dataset D3. The red line shows the point where the shift has occurred.

Method A is the learning without CSD, and the lowest classification error is achieved from dataset D1 for the three base classifiers, because there is only one class covariate shift. Other datasets have high classification error because there is no adaptation in these methods and due to the amount of shifts in the data. Figure 5(a) shows the classification error graph, where the classification error increases after the shift has occurred. This shows that the traditional learning algorithm is not suitable for non-stationary data.

Method B is an adaptive learning with CSD and retraining the classifier on initial KB_0 plus new knowledge KB_{New} obtained from the delay in shift-detection. In this method, the KB_{New} has a low impact on the $KB_{Updated}$ because it contains only a window of new information obtained from the validation period of the shift-detection procedure. The results from the predicted labels are very closely related to the supervised labels. For all the classifiers and datasets, the classification error for the supervised labels are only slightly better, which shows that method B has not performed worse than the traditional learning.

Method C is also an adaptive learning with CSD and retrains the classifier only on the new predicted information KB_{New} after the shift-detection. The KB_{New} is the data from current distribution. During the classification in testing phase after the shift-detection and before the adaptation, the chance of getting the correct label from the classifier depends upon the

amount of shift. The shaded area in Fig 3 is the region where the chance of getting wrong labels is high; hence the data belonging to this area will be predicted wrongly. But, still the classification from unshaded area are correctly made, so retraining upon that may lead to improved classification performance according to the results presented in tables above. For datasets D4 and D6, where altering and stairs of covariate shifts are present, the classification accuracies are much better compared to other data-sets. This shows that this approach is good for dealing with non-stationary environments where the covariate shift has affected the data. Figure 5(b) shows the classification error graph, where the classification error decreases after the shift-detection. Moreover, the performance of predicted labels are closest to the supervised labels. Hence, this method suits well with learning in non-stationary environments.

Method D is a learning with CSD, similar to the method B, but here the data from initial KB_0 is selected based on the Euclidean distance from the mean of KB_{New} to the KB_0. The performance of this approach lies somewhere in the middle of all compared methods. This method has not shown a drastic improvement in non-stationary learning but it is better than traditional learning.

Method E is a transductive learning based approach, where on each shift detection, the data around the current input is considered. The nearest neighbours in KB_0 from current input are selected and combined with KB_{New}. The performance of this approach is the second best because, the input data is from the shifted distribution and the half of the supervised information is used from the old distribution. So, mixture of transductive approach and new knowledge is good for learning in the non-stationary environments. Here also, it can be seen that performance of predicted labels are second best after the method C.

One of the most important conclusions from the results is that the classification error of the classifier trained on predicted labels *Pred* is comparable to that of the classifier trained on supervised labels *Sup*. This is important as in reality we usually don't have the actual labels online and the results demonstrate that the proposed method works well using predicted labels, specifically for the methods C and E.

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 shift-detection test plays an important role in initiating a correct adaptive action. The window size of the shift-detection at second stage of validation plays a crucial role in adaptation. In shift-detection we have used a heuristic approach in choosing the size of the window. Based on the window size, the adaptation is performed on the combinations of predicted labels and initial knowledge base. Although detailed mathematical analysis of the window size and the adaptation methods are yet to be performed, through the experimental results and discussion, it can be easily seen that the proposed adaptive learning methods with shift-detection test suits well for learning in non-stationary environments.

VII. Conclusion

The proposed ALCSD algorithm is a flexible tool for non-stationary learning and dealing with covariate shift in the input data distribution. In this paper, several methods are proposed for covariate shift-adaptation using a two-stage covariate shift detection test (CDT) involving an EWMA control chart in the first stage and Kolmogorov–Smirnov hypothesis test in the second stage. The CDT is a useful test to detect the covariate non-stationary shifts and drift in the data. Based on the detected shifts, the algorithm initiates adaptive action. The performance of the suggested methods show good results in terms of low classification errors on re-training the classifiers based on new predicted labels. This algorithm is computationally efficient because it does not require an additional memory to store all the observations. Only, the initial knowledge-base (KB_0) and a small window of KB_{New} are to be kept. Experimental analysis shows that the performance of the approach is good in a range of non-stationary situations with various types of covariate shift over the traditional learning approach. This work is planned to be extended further by employing it into real-world problems involving multivariate data.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[2] R. O. Duda, P. E. Hart, and D. G. Stork., *Pattern Recognition*. Wiley-Interscience, 2001.

[3] S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks*, vol. 1, no. 1, pp. 17–61, Jan. 1988.

[4] M. G. Kelly, D. J. Hand, and N. M. Adams, "The Impact of Changing Populations on Classifier Performance," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1998, vol. 32, no. 2, pp. 367–371.

[5] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[6] V. Vapnik, "An Overview of Statistical Learning Theory," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 988–99, Jan. 1999.

[7] G. Widmer and M. Kubat, "Learning in the Presence of Concept Drift and Hidden Contexts," *Mach. Learn.*, vol. 101, no. 23, pp. 69–101, 1996.

[8] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with Drift Detection," in *Advances in Artificial Intelligence–SBIA 2004.*, 2004, pp. 286–295.

[9] A. Bifet and R. Gavalda, "Learning from Time-Changing Data with Adaptive Windowing," *SDM*, 2007.

[10] C. Alippi and M. Roveri, "Just-in-Time Adaptive Classifier--Part II: Designing the Classifier," *IEEE Trans. Neural Networks*, vol. 19, no. 12, pp. 2053–2064, 2008.

[11] C. Alippi, G. Boracchi, and M. Roveri, "A Just-In-Time Adaptive Classification System Based on the Intersection of Confidence Intervals Rule," *Neural Networks*, vol. 24, no. 8, pp. 791–800, Oct. 2011.

[12] C. Alippi, G. Boracchi, and M. Roveri, "Just-In-Time Ensemble of Classifiers," in *Neural Networks (IJCNN), The 2012 International Joint Conference on. IEEE*, 2012, pp. 1–8.

[13] 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.

[14] C. Alippi, G. Boracchi, and M. Roveri, "Just-In-Time Classifiers for Recurrent Concepts," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 4, pp. 620–634, Apr. 2013.

[15] K. B. Dyer, R. Capo, R. Polikar, and S. Member, "COMPOSE□: A Semisupervised Learning Framework for Initially Labeled Nonstationary Streaming Data," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–15, 2013.

[16] 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.

[17] 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.

[18] A. J. Storkey, "When Training and Test Sets are Different□: Characterising Learning Transfer," in in *Dataset Shift in Machine Learning*, no. Section 12, Press, MIT, 2010, pp. 3–28.

[19] M. J. G. Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A Unifying View on Dataset Shift in Classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, Jan. 2012.

[20] M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[21] K. Wang, C. A. Fu, and J. X. Yu, "Mining Changes of Classification by Correspondence Tracing," in *In Proceedings of the 3rd SIAM International Conference on Data Mining (SDM-03)*, 2003, pp. pp. 95–106.

[22] N. Japkowicz, "Assessing the Impact of Changing Environments on Classifier Performance," in *The Proceedings of the 21st Canadian Conference in Artificial Intelligence*, 2008, pp. 13–24.

[23] Y. Yang, X. Wu, and X. Zhu, "Conceptual Equivalence for Contrast Mining in Classification Learning," *Data Knowl. Eng.*, vol. 67, no. 3, pp. 413–429, Dec. 2008.

[24] D. a. Cieslak and N. V. Chawla, "A Framework for Monitoring Classifiers' Performance: When and Why Failure Occurs?," *Knowl. Inf. Syst.*, vol. 18, no. 1, pp. 83–108, May 2008.

[25] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data," *Mach. Learn.*, vol. 1, no. 3, pp. 317–354, 1986.

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