Just-In-Time Ensemble of Classifiers

Cesare Alippi, Giacomo Boracchi, Manuel Roveri
Politecnico di Milano, Dipartimento di Elettronica e Informazione, Milano, Italy
{alippi,boracchi,roveri}@elet.polimi.it

***Abstract*—Handling dynamic environments and building up algorithms operating at low supervised-sample rates are two main challenges for classification systems designed to operate in real-life scenarios. Here, changes in the probability density function of classes characterizing the data-generating process (also called concept drift) should be detected as soon as possible to prevent the classifier from becoming obsolete. Moreover, when the rate of supervised samples during the operational life is low (as in those situations where the sample inspection is costly or destructive) both detecting the change and re-training the classifier become even more critical aspects.**

We present an adaptive classifier that exploits both supervised and unsupervised data to monitor the process stationarity. The classifier follows the just-in-time (JIT) approach and relies on two different change-detection tests (CDTs) to reveal changes in the environment and reconfigure the classifier accordingly. The proposed solution assesses the stationary in both the joint probability density function (CDT at the classification error) and the distribution of the inputs (CDT on unlabeled data). In addition, we integrate in the JIT adaptive classifier a procedure able to handle recurrent concepts within an ensemble of classifiers framework. Experiments show that monitoring unsupervised samples and handling recurrent concepts is essential for classifying in non-stationary environments when few supervised samples are available.

***Index Terms*—Concept drift, adaptive classifiers, low supervised-data rates, recurrent concepts.**

I. Introduction

Classification applications where the probability density function (pdf) of the data-generating process evolve over time are referred as *concept drift*. Relevant examples of concept drift are the abrupt and gradual concept drift cases. Abrupt concept drift refers to situations where the process suddenly changes from a stationary state to another stationary state, e.g., due to a permanent or a transient fault. Differently, gradual concept drift refers to cases where the process continuously changes over time, a situation typically caused by aging effects or thermal drift.

Classification systems designed to deal with concept drift, e.g., [1], have to operate when the data-generating process is partially known, taking into account that they might become obsolete and lose their effectiveness, when the process changes. To preserve their accuracy, these classifiers must be able to adapt and react to changes (non-stationary conditions). In this direction, these classifiers must guarantee effective change-detection abilities to distinguish between changes in the data-generating process and noise (which may induce false positives), should increase the classification accuracy during the operational life in stationary conditions (by exploiting possible supervised information available), and should be possibly endowed with mechanisms to successfully address recurrent concepts (i.e., situations in which the process turns into an already explored state).

Though different approaches for concept drift have been presented in the literature (e.g., instance weighting, instance selection, ensemble of classifiers) the main leitmotiv behind these solutions is their ability to detect a change and react accordingly. Change detection is thus an essential step to make the classification system adaptive to concept drift. Two main approaches for change detection in classification systems have been proposed in the related literature, which differ for the analyzed entity: the classification error or the input observations.

The former approach (e.g., [2]–[4]) aims at monitoring changes in the joint pdf by evaluating variations in the classification error computed on supervised data. The classification system presented in [3] relies on a fixed threshold on the classification accuracy, while [2] relies on an adaptive thresholding mechanism determined by the confidence interval on a reference minimum accuracy. In both cases, a change is detected as soon as the classifier's accuracy falls below a threshold. The solution presented in [4] relies on an exponentially weighted moving average chart applied to the classification error.

The latter approach assesses the non-stationarity of the probability density function (pdf) of the inputs by monitoring all the observations disregarding their label values. The Just-In-Time (JIT) classifiers presented in [5]–[8] follow this approach.

Monitoring the classification error allows the classifier for reacting to changes when these directly influence its accuracy, and this is in principle preferable. However, assessing the stationarity by means of the classification error becomes critical in applications where obtaining supervised information is difficult or costly. Then, a viable option at low supervised-sample rates consists in monitoring the distribution of unlabeled observations. Unfortunately, this solution does not allow one to detect changes that do not affect the distribution of observations, even when these determine a dramatic fall in the classifier's accuracy (e.g., the swap of the classes). It has also to be mentioned that monitoring changes in the classification error can be performed both on quantitative or qualitative observations, while monitoring the unlabeled data typically requires quantitative observations.

The adaptation phase that follows each concept-drift detection is typically piloted by the classification error. The classification error allows one to identify the obsolete knowledge base to be removed [9], aggregate an ensemble of classifiers

according to the estimated accuracy [10], [11] or reactivate previously trained classifiers [12]. Also in these cases, the main drawback is the need of a sufficient amount of supervised samples to reliably assess the classification error. Once again, the unlabeled observations could be considered to identify within the most recent observations, those that are up-to-date with the current state of the data-generating process, thus representing an ideal candidate for constituting the new knowledge base [8]. However, since analyzing the unlabeled observations does not allow one to perceive changes that do not affect the distribution of observations (e.g., the swap of the classes), it is necessary to consider additional information to successfully deal with recurrent concepts.

In this paper we present an effective solution for exploiting the classification error within the JIT framework. This novel solution allows us to increase the effectiveness of the change-detection phase by monitoring both the distribution of the observations and the classification error. Thus, the JIT classifier exploits both supervised and unsupervised samples to adapt to changes in the data-generating process and, de facto, the proposed solution extends the JIT framework to deal with recurrent concepts. Every time a concept drift is detected, the previously trained classifiers are tested to identify if the current concept has been already envisaged or not. If the concept is recurrent, the previous classifier is re-activated (together with the new knowledge); otherwise a new classifier is introduced.

The experiments allow us to investigate in practice how the amount of supervised information influences the classifier accuracy in nonstationary data-generating processes. Results are coherent with the intuitive idea that exploiting unsupervised data and handling recurrent concepts become essential elements for learning in non stationary environments when the amount of supervised samples is scarce.

The novel contributions of the paper can be summarizes as:

1) We introduce a novel JIT classifier that exploits both the supervised and unsupervised samples to effectively adapt to concept drift. The proposed adaptive classifier relies on two different change-detection tests (CDTs) to assess the stationarity of the data-generating process: the former CDT is meant for monitoring the stationarity of the observations disregarding their label (CDT_X), while the latter assesses the change if the classification error (computed on supervised samples) is stationary (CDT_ϵ). This approach is particularly promising in case of low supervised-sample rates as it exploits the promptness of CDT_X, while maintaining the detection ability also in non-stationarity cases that do not affect the distribution of the observations.
2) A specific solution for CDT_ϵ designed for assessing if the classification error is stationary: CDT_ϵ operates on Bernoulli sequences, which reliably model the classification error measured at supervised samples. The proposed CDT is based on the ICI-based CDT [8], [13].
3) A procedure to handle recurrent concepts within an ensemble of classifiers framework. The classifier is retrained using the knowledge base previously acquired that is coherent with the current concept. This procedure allows the classifier to compensate the shortage of supervised samples by reactivating the knowledge base already acquired whenever this is coherent with the current state of the process.

The paper is organized as follows: Section II states the problem and introduces the formalism used; Section III gives an outline of the proposed JIT adaptive classifier and discusses in details the core techniques such as the change-detection test on classification error and the procedure to handle recurrent concepts. Section IV details the complete algorithm, while experimental results are presented in Section V.

II. Problem statement

Let us consider the concept drift framework in which the input samples (observations) are scalar entities generated from process X according to an unknown distribution. Denote by $x_t \in \mathbb{R}$ the observation at time t, and by y_t the class label associated with x_t. In what follows, without loss of generality, we consider a two-class classification problem, i.e., $y_t \in \{\omega_1, \omega_2\}$. The probability density function of the inputs at time t can thus be defined as

$$p(x|t) = p(\omega_1|t)p(x|\omega_1, t) + p(\omega_2|t)p(x|\omega_2, t), \qquad (1)$$

where $p(\omega_1|t)$, $p(\omega_2|t) = 1 - p(\omega_1|t)$ are the probabilities of getting a sample of class ω_1 and ω_2, respectively, while $p(x|\omega_1, t)$, $p(x|\omega_2, t)$ are the conditional probability distributions at time t. Both the probabilities of the classes and the conditional pdfs are assumed to be unknown and may evolve over time, whenever a non-stationarity occurs.

The training sequence consists in the first T_0 observations that are assumed to be generated in stationary conditions, i.e., $p(\omega_1|t)$, $p(\omega_2|t)$, and $p(x|\omega_1, t)$, $p(x|\omega_2, t)$ do not change within the time interval $[0, T_0]$. Supervised pairs (x_t, y_t) are provided both within the training sequence and during the operational life (i.e., $t > T_0$). However, no assumption is made on how often these supervised pairs are provided, as these could be received following a regular time-pattern (e.g., one supervised sample out of m) or even intermittently.

III. The proposed solution

The main characteristic of the proposed JIT classifier is the integration of a CDT on the classification error for monitoring the stationary of a data-generating process. This improves the change-detection abilities of JIT adaptive classifiers suggested in [6], [8], [14]–[16] relying on a single CDT to monitor the stationarity of the distribution of X, disregarding the existence of supervised labels.

The key elements of the proposed solution are:

CDT_X: the CDT that analyzes the raw observations to monitor the stationarity of x_t, disregarding their labels;

CDT_ϵ: the CDT for assessing if the average classification error changes over time (CDT_ϵ operates only on supervised samples);

K: the classifier used to classify input samples;

```
1-  Configure K, CDT_X and CDT_ε from the training sequence;
2-  while (1) do
3-      input receive new data x_t;
4-      if (Either CDT_X or CDT_ε detects a nonstationarity at time t) then
5-          Characterize the current concept C_i;
6-          Check if C_i is coherent with any of {C_j, j < i};
7-          Flush the knowledge base from K;
8-          if (C_i is recurrent) then
9-              K is trained on the training sequences of the previous occurrences of C_i;
10-             Integrate C_i;
            else
11-             Configure and activate K from C_i;
            end
12-         Configure both CDT_X and CDT_ε on C_i;
        end
13-     if (Supervised label y_t is provided) then
14-         Integrate (x_t, y_t) in the knowledge-base of K;
15-         Update K;
        else
16-         Assign the label K(x_t) to x_t.
        end
    end
```

Algorithm 1: High-level description of the proposed JIT adaptive classifier.

C_i: the i-th concept, which has to be considered as a set of observations (together with supervised labels when available) associated to a specific state of the data-generating process.

The proposed solution is described in Algorithm 1. After a preliminary configuration phase (line 1), both CDT_X and CDT_ϵ are used to assess the process stationarity (typically they operate on a window of data and, therefore, stationarity is assessed at window level). As soon CDT_X or CDT_ϵ detects a change, the current concept C_i is identified (line 5) by extracting a subset of observations (both supervised and unsupervised, details are provided in Sections III-B and III-C) that represent the current state of the process. These observation are used to train the proposed JIT adaptive classifier on the concept C_i. Afterwards, the knowledge base of K is removed (line 7) and then K is reconfigured by using the updated knowledge base from C_i, as well as possible previously acquired training samples whenever C_i is recurrent. To identify recurrent concepts, C_i is compared with concepts previously encountered $\{C_j, 0 \leq j < i\}$ using the procedure described in Section III-C (line 6). When C_i is recurrent, the classifier K is reconfigured using all the training samples belonging to concepts C_j that are recognized as previous occurrence of C_i (line 9), and from the recently identified training sequence (line 10). Both CDT_X and CDT_ϵ are configured from the training sequence referred to C_i. Inputs are then classified using K (line 16), which is updated every time a supervised information is provided (line 15). Note that only one concept at a time is active (i.e., used to train K): however, the training sequences from different concepts remain stored for the next need.

Details concerning CDT_ϵ are discussed in Section III-A, while the recurrent concept analysis is discussed in Sections III-C. The complete algorithm is exhaustively described in Section IV.

A. The ICI-based CDT for Bernoulli Sequences

The proposed JIT adaptive classifier exploits the ICI-based CDT [8] as CDT_X, for its effectiveness in detecting both abrupt or gradual concept drift (in terms of low false positive and negative rates as well as low detection delays) and for its reduced computational complexity. Like other CDTs exploiting the ICI-rule, this CDT relies on specific features characterizing the distribution of X which, in stationary conditions, are i.i.d. and follow a Gaussian distribution. The features are extracted from disjoint subsequences of data, and are derived from the sample mean and the sample variance computed on data subsequences (the latter follows the Gaussian distribution thanks to and ad-hoc transformation [17]). Then, the ICI-rule [18] can be used to assess, on-line and sequentially, if the feature values have been generated from the same Gaussian distribution. The configuration of CDT_X consists in extracting the feature values from the training sequence and computing a confidence interval for the expected value of each feature.

The proposed CDT_ϵ consists in a customization of the ICI-based CDT to assess if the classification error is constant, relying on the fact that classification errors can be modeled as i.i.d. realizations of a Bernoulli random variable. During the JIT training phase we configure an additional classifier K_0 exclusively for change-detection purposes which, during the operational life, measures the classification error on supervised samples. For this reason, K_0 is not updated whenever supervised samples are provided, thus guaranteeing that its classification error – in stationary conditions – is constant (in statistical sense). The classifier K_0 has not to be confused with K, which associates a label to each input sample.

Let (x_t, y_t) be a supervised couple and let $K_0(x_t)$ be the label that classifier K_0 associates to x_t. We denote the element-wise classification error of K_0 as

$$\epsilon_t = \begin{cases} 0, & \text{if } y_t = K_0(x_t); \\ 1, & \text{otherwise,} \end{cases} \tag{2}$$

which can be modeled over time as sequence of i.i.d. Bernoulli random variables whose parameter p_0 represents the expected classification error of K_0. In stationary conditions, p_0 has to be constant since K_0 is not updated. We assess the non-stationarity (both in the classes' probability or in the conditional pdfs) using the average classification error computed on disjoint subsequences of ν supervised observations. Since the sum of ν Bernoulli random variables follows a Binomial distribution $\mathcal{B}(p_0, \nu)$, it can be approximated by a Gaussian

distribution whenever ν is sufficiently large, i.e.,

$$\mathcal{B}(p_0, \nu) \sim \mathcal{N}\big(p_0\nu, p_0(1-p_0)\nu\big). \quad (3)$$

Thanks to this approximation, we can directly apply the ICI-rule to sequentially assess if the average error of K_0, computed on non-overlapping sequences of ν supervised samples, is constant over time.

The configuration step in CDT$_\epsilon$ differs from that of CDT$_X$ since both the mean and variance of the Gaussian distribution (3) is determined by p_0 and ν, thus it is not necessary to compute the sample standard deviation of the average error of K_0. It follows that ν supervised samples are enough to configure CDT$_\epsilon$. In our implementation K_0 is a *k*-NN classifier and we split the supervised samples belonging to the training sequence $[1, T_0]$ in two subsets TS_0 and VS_0: TS_0 is used to train K_0, while the classification error of K_0 is measured on VS_0 and is used to train CDT$_\epsilon$.

Finally, both CDT$_X$ and CDT$_\epsilon$ operate on subsequences of data, processing them asynchronously since the arrival of supervised couples is not assumed to be uniform and could even be sparse.

B. Change Validation

According to the JIT approach, the knowledge base of the classifier has to be renewed at each non-stationarity detection. This strategy aims at guaranteeing that the classifier relies only the up-to-date knowledge base. On the other hand, false positives (i.e., detections that do not correspond to an actual change in the distribution of X or in the classification error of K_0) result in a loss of precious supervised information, which could not be bearable when the supervised information is scarce. To this purpose, we introduce in the JIT classifier a change-validation procedure following the approach that yielded the hierarchical ICI-based CDT [14], which is here extended to validate changes on both the distribution of the data-generating process and the classification error.

The change-validation procedures exploit two sets of observations that are considered to be generated before and after the concept drift. To this purpose, the CDTs relying on the ICI-rule turn to be particularly useful as these are naturally endowed with a refinement procedure [8] providing, after each detection, an estimate of the time-instant the change occurred. Whenever either CDT$_X$ or CDT$_\epsilon$ detects a non stationarity at time $\hat{T}$, the corresponding refinement procedure is executed providing an estimate T_{ref}: the observations at time instants $[0, T_0]$, representing the process in its original state, are compared with those at time instants $[T_{\text{ref}}, \hat{T}]$ that represent the process after the detection (to be validated).

The change-validation procedure on the data (CVP$_X$) corresponds to the second level CDT of the hierarchical ICI-based CDT [14] and it relies on the features computed by CDT$_X$, which follow a Gaussian distribution and, therefore, the change-validation problem can be formulated as a multivariate hypothesis test, using the Hotelling T^2 statistics [19]. More specifically, we stack in column vectors (each row representing a feature) the features extracted form the observations in $[0, T_0]$ and in $[T_{\text{ref}}, \hat{T}]$, we compute their sample means on each of these two sets, namely $\overline{F_0}$ and $\overline{F_1}$, and their pooled sample covariance. We can then formulate the null hypothesis "$\overline{F_0} - \overline{F_1} = 0$" and do inference using the Hotelling T^2 statistic such that the null hypothesis can be rejected according to a predefined confidence level α.

Similarly, we reformulate the change-validation procedure on the classification error (CVP$_\epsilon$) as an inference problem on the proportions of two populations. In this case the problem becomes univariate and the null hypothesis is $\bar{\epsilon}_0 - \bar{\epsilon}_1 = 0$, being $\bar{\epsilon}_0$ and $\bar{\epsilon}_1$ the average classification error of K_0 computed on observations provided within $[0, T_0]$ and $[T_{\text{ref}}, \hat{T}]$, respectively. The test statistic follows a Gaussian distribution, which can be rejected according to a defined significance level α (that in principle could differ from that used in CVP$_X$).

Any detection has to be validated, and to this purpose both validation procedures CVP$_X$ and CVP$_\epsilon$ are executed, disregarding which CDT rose the non-stationarity flag. The classifier K is reconfigured (Algorithm 1 line 5) when at least one validation is confirmed, otherwise the change is discarded and only the CDT providing the false detection is reconfigured, while the classifier K and its knowledge base are not modified (details are presented in Section IV, Algorithm 2).

C. Identifying Recurrent Concepts

Recurrent concepts are identified by testing both observations and classification error simultaneously. The refinement procedures of CDT$_X$ and CDT$_\epsilon$ provide a subset of observations that, when the change is validated, is coherent with the current state of the process. Thus, C_i, i.e., the concept characterizing the non-stationarity detected at $\hat{T}_i$, can be handled by means of the observations within $[T_{\text{ref},i}, \hat{T}_i]$ (both supervised and unsupervised), being $T_{\text{ref},i}$ the output of the corresponding refinement procedure.

We store each concept C_i, which corresponds to both supervised and unsupervised observations within $[T_{\text{ref},i}, \hat{T}_i]$, to possibly recover it and hence deal with recurrent concepts. Recurrent concepts are identified by directly comparing the current concept C_i with all the previous concepts $\{C_j, 0 \leq j < i\}$ (C_0 is the concept representing the initial training sequence): concept comparison is performed by analyzing the quantities analyzed by the change-validation procedures (described in Section III-B).

More specifically, when comparing two concepts C_i and C_j, $i \neq j$, we assess if both the average of the features and the classification errors computed in $[T_{\text{ref},i}, \hat{T}_i]$ correspond to those computed in $[T_{\text{ref},j}, \hat{T}_j]$. To this purpose, we enforce $\overline{F_i}$, $\overline{F_j}$, $\bar{\epsilon}_i$ and $\bar{\epsilon}_j$, which have been computed during the validation procedures, and test if their difference falls below a used-defined threshold. Such an assessment is performed by means of the following thresholding:

$$\begin{cases} \frac{\|\overline{F_i} - \overline{F_j}\|_2}{\|\overline{F_i}\|_2 + \|\overline{F_j}\|_2} < \gamma, \\ \\ \frac{|\bar{\epsilon}_i - \bar{\epsilon}_j|}{\bar{\epsilon}_i + \bar{\epsilon}_j} < \gamma \end{cases}, \quad (4)$$

where $\gamma \in [0, 1]$ is a tuning parameter that determines to which extent two concepts having similar features and classification errors should be considered recurrent.

When C_i and C_j , $j \neq i$ satisfy both the above conditions we consider the concepts C_i to be recurrent, and the supervised samples in $[T_{\text{ref},j}, \hat{T}_j]$ can be safely paired with those in $[T_{\text{ref},i}, \hat{T}_i]$ to configure K.

We are aware that (4) is a rather naive solution as the difference between features (classification errors) is measured relatively to their norm: as such features (classification errors) very close could be considered different when they are too small. We are currently investigating more effective solutions to handle recurrent concepts by analyzing simultaneously the observations and the classification errors.

D. JIT Reconfiguration

When testing if C_i is a recurrent concept, it may happen that, among all the previous concepts $\{C_j, 0 \leq j < i\}$, more than a concept satisfies (4). This corresponds to a concept that occurs more than once: in these situations we can configure K by exploiting all the supervised samples from the previous occurrences of C_i. In general, the classifier K at the i-th non-stationarity $\hat{T}_i$ is configured from all the supervised couples $Z = \{(x_t, y_t), t \in I\}$, where

$$I = [T_{\text{ref},i}, \hat{T}_i] \cup \bigcup_{j | C_i \text{ and } C_j \text{ satifsy (4)}} [T_{\text{ref},j}, \hat{T}_j]\,. \qquad (5)$$

Recurrent concepts are not used to reconfigure the CDTs empowered in the JIT adaptive classifier, as these can be rather successfully configured from most recent samples, without the need to include additional observations from the recurrent concepts. In fact, the change-validation procedures typically requires a minimum size for $[T_{\text{ref}}, \hat{T}]$ which allows a proper configuration of the CDTs.

IV. Algorithm Details

We adopt the following notation: I refers to a set of time instants t in which the observations arrived (as in (5)), O stands for the set of observations x_t (both supervised and unsupervised), and Z contains only the supervised pairs (x_t, y_t). To ease the notation, we associate to each concept C_i the time interval $[T_{\text{ref},i}, \hat{T}_i]$ which in practice is used to identify recurrent concepts (i.e., Eq. 4).

The proposed solution –detailed in Algorithm 2– follows the outlines of JIT adaptive classifier of Algorithm 1 and, in our implementation, we exploit a *k*-NN classifier for both K and K_0 thanks to their easy update and re-configuration step.

During the configuration phase of the JIT adaptive classifier, the supervised samples in the training sequence are split into TS_0 and VS_0 (line 3), to train K_0 on TS_0 (line 4), and configure CDT_ϵ on the classification errors of K_0 computed over VS_0. All the observations in the training set are used to configure CDT_X (line 6) and all the supervised information to train K (line 4).

During the operational life, possible supervised samples are integrated in the current knowledge base (lines 11 - 12) to update or reconfigure the classifier K (line 13). In the specific case of k-NN, the update step consists in including the new supervised samples in the knowledge base and updating the k parameter as described in [6]. Supervised information are labelled by K_0 and the error $|K_0(x_t) - y_t|$ is used by CDT_ϵ to assess the stationarity of the classification error. Each observation x_t is also used to assess the stationarity of X by CDT_X (line 15). It has to be stressed that CDT_X and CDT_ϵ operate, possibly asynchronously, on disjoint subsequences of data, thus, the CDT output is not provided at each sample.

Whenever a detection occurs the refinement procedure [8] is executed to determine T_{ref}, which allows to validate the detection by using both the change-validation procedures CVP_X and CVP_ϵ (line 19): if any of the two procedures validates the detection (see Section III-B), the change is confirmed. Each validated non-stationarity gives raise to a concept C_i, which is stored in the concept library, and compared with the concepts previously encountered (line 23) by using the procedures described in Section III-C and (4).

After each non-stationarity detection, the knowledge-base of the classifier K is created by merging all supervised samples from compatible concepts as in (5). Both CDTs are instead trained without resorting to previous observations (line 29 - 30), see Section III-D.

Finally, the unsupervised observations x_t are classified by using the up-to-date classifier K.

V. Experiments

To show the substantial improvements achievable by combining the CDT on the classification error and on the observations, we compare the proposed JIT adaptive classifier with the JIT adaptive classifiers relying on CDT_X and on CDT_ϵ only. Such experiments aim also at investigating to which extent the amount of supervised information provided during the operational life affects the CDTs and, hence, the JIT adaptive classifiers accuracy. This experimental analysis provides useful guidelines to design adaptive classifiers depending on the amount of supervised samples available during the operational life.

A. Dataset Generation

We synthetically generate datasets of $N = 20000$ observations according to five different data-generating processes, which have been reported in Table I (classes have the same probability, i.e., $p(\omega_1) = p(\omega_2)$. We consider five scenarios:

- *test ID 1:* an abrupt concept drift affecting both classes, i.e., at time $t = 10000$ the concept moves from C_1 to C_4;
- *test ID 2:* abrupt concept drift resulting in a classes' swap, i.e., at time $t = 10000$ the concept C_1 becomes C_3;
- *test ID 3:* an abrupt concept drift affecting $p(x|\omega_1)$, i.e., at $t = 10000$ the concept shifts from C_1 to C_2;
- *test ID 4:* a transient concept drift, where the concept changes from C_1 to C_4 at $t = 5000$ and then returns in C_1 at $t = 10000$;

1- $I = \{1, \ldots, T_0\}$;
2- $O = \{x_t,\ t \in I\}$, $Z = \{(x_t, y_t),\ t \in I\}$;
3- Partition Z into TS_0 and VS_0;
4- Train K on Z and K_0 on TS_0;
5- Configure CDT$_\epsilon$ using K_0 on VS_0;
6- Configure CDT$_X$ using O;
7- $C_0 = [1, T_0]$;
8- $t = T_0 + 1$; $i = 1$;
9- **while** *(x_t arrives)* **do**
10- **if** *(y_t is provided)* **then**
11- $I = I \cup \{t\}$;
12- $Z = Z \cup \{(x_t, y_t)\}$;
13- update / retrain K on Z;
14- run CDT$_\epsilon$ on $|y_t - K_0(x_t)|$;
end
15- Apply CDT$_X$ on x_t;
16- **if** *(CDT$_\epsilon$ or CDT$_X$ detects a non-stationarity)* **then**
17- $\hat{T} = T$;
18- Estimate T_{ref} with the ICI-based refinement procedure ([8]);
19- **if** *(change is validated by CVP$_X$ or CVP$_\epsilon$)* **then**
20- $T_{\text{ref},i} = T_{\text{ref}}$, $\hat{T}_i = \hat{T}$;
21- $C_i = [T_{\text{ref},i}, \hat{T}_i]$;
22- $I = [T_{\text{ref},i}, \hat{T}_i]$;
23- **for** *(j = 0 ; j < i ; j++)* **do**
24- compare C_i and C_j using (4);
25- **if** *(C_i is an occurrence of C_j)* **then**
26- $I = I \cup [T_{\text{ref},j}, \hat{T}_j]$;
end
end
27- $O = \{x_t,\ t \in [T_{\text{ref},i}, \hat{T}_i]\}$;
28- $Z = \{(x_t, y_t),\ t \in I\}$;
29- train K on Z and configure CDT$_X$ on O;
30- configure CDT$_\epsilon$ using K_0 on $\{(x_t, y_t),\ t \in [T_{\text{ref},i}, \hat{T}_i]\}$;
31- i++;
end
end
32- **if** *(y_t is NOT provided)* **then**
33- compute $\hat{y}_t = K(x_t)$;
end
end

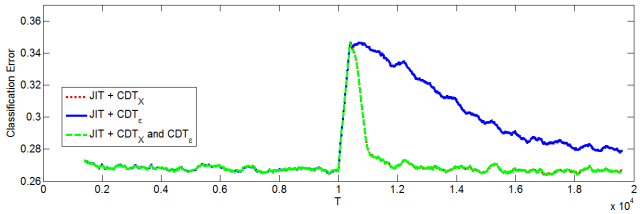
Algorithm 2: The proposed JIT Adaptive Classifier in details

- *test ID 5:* a sequence of abrupt concept drifts occurring every 4000 observations and shifting the concept from C_1 to C_4 and vice-versa.

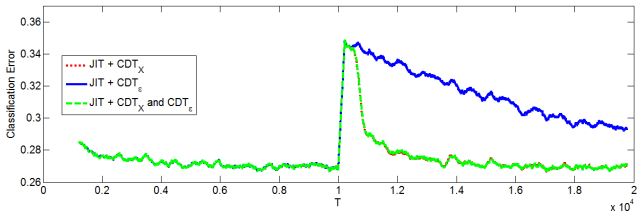
To ease the comparison we assume that supervised samples are periodically distributed (one supervised pair is provided every m observations) and each dataset is analyzed assuming different values of $m = 2, 10, 20$. The initial training sequence is composed of $T_0 = 1000$ observations, with supervised pairs provided every m observations.

TABLE I
CONSIDERED CONCEPTS

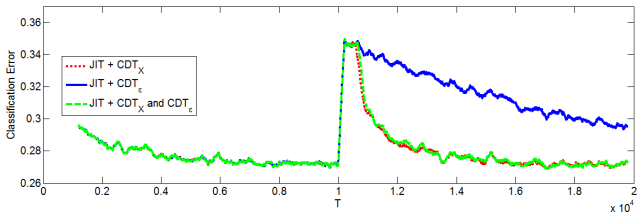
Concept	$p(x\|\omega_1)$	$p(x\|\omega_2)$
C_1	$\mathcal{N}(0, 4)$	$\mathcal{N}(2.5, 4)$
C_2	$\mathcal{N}(0, 4)$	$\mathcal{N}(4.5, 4)$
C_3	$\mathcal{N}(2.5, 4)$	$\mathcal{N}(0, 4)$
C_4	$\mathcal{N}(2, 4)$	$\mathcal{N}(4.5, 4)$



(a) $m = 2$



(b) $m = 10$

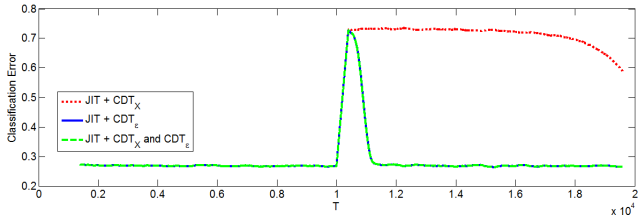


(c) $m = 20$

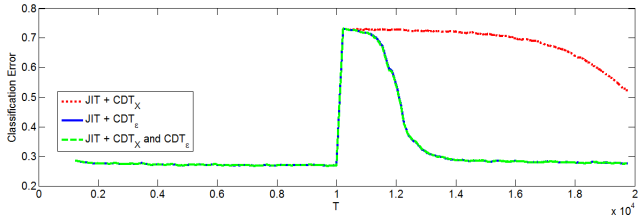
Fig. 1. Test ID 1: comparison of the classification errors as function of time for the proposed JIT adaptive classifier and the solutions based on CDT$_X$ and on CDT$_\epsilon$ only.

Both CDT$_X$ and CDT$_\epsilon$ have been configured following the guidelines presented in [8] for ICI-based CDT, and in particular we set $\nu = 20$ in both CDTs; the classifier K_0 has been trained on the supervised samples within $[1, T_0/2]$, and CDT$_\epsilon$ has been configured on the classification errors of K_0 computed from the remaining supervised samples, i.e., $[T_0/2+, T_0]$. Both K and K_0 are k-NN classifier and their parameter k is estimated via cross-validation following the procedure detailed in [6]. Finally, in (4) we set $\gamma = 0.3$. Classification errors along the datasets are plotted in Fig. 1 - Fig. 5, where it is shown the classification errors (for the three different JIT adaptive classifiers) averaged over 500 dataset realizations and smoothed over the previous 200 observations.

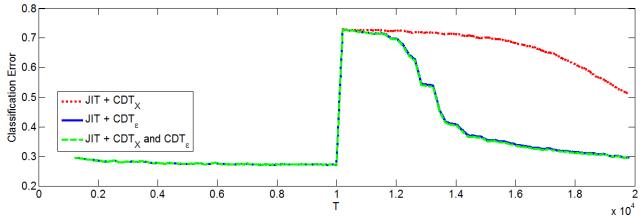
Fig. 1 shows the experimental results for *test ID 1*. These results are particularly meaningful to analyze the effectiveness of the proposed solution: when concept drift occurs, the classification error increases for all the classifiers but the CDT$_X$ detects the change more promptly than the others. Thus, both the proposed JIT (dashed line) and the JIT based on CDT$_X$ (dotted line) promptly react to the concept drift (i.e.,



(a) $m = 2$

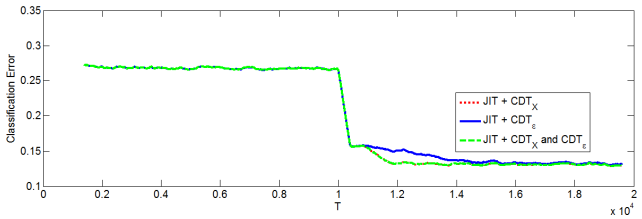


(b) $m = 10$

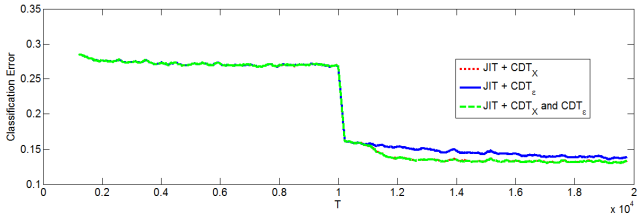


(c) $m = 20$

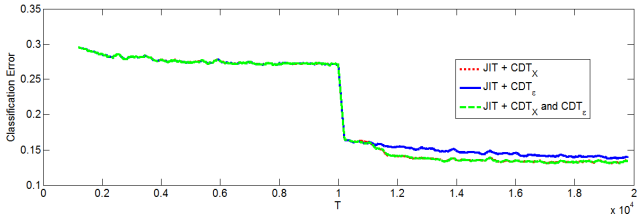
Fig. 2. Test ID 2: comparison of the classification errors as function of time for the proposed JIT adaptive classifier and the solutions based on CDT_X and on CDT_ϵ only.



(a) $m = 2$



(b) $m = 10$



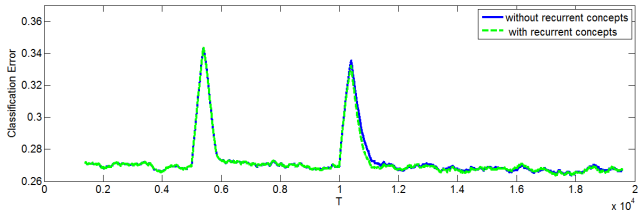
(c) $m = 20$

Fig. 3. Test ID 3: comparison of the classification errors as function of time for the proposed JIT adaptive classifier and the solutions based on CDT_X and on CDT_ϵ only..

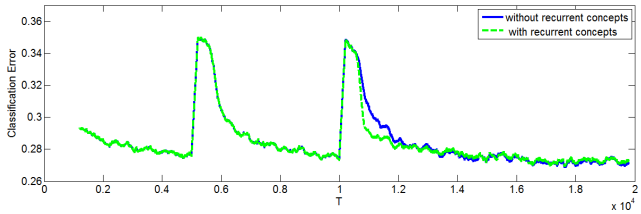
the classification error decreases). In contrast, CDT_ϵ shows longer detection delays than CDT_X: hence, when the concept drift occurs the JIT based on the CDT_ϵ (solid line) takes longer to achieve back the previous accuracy levels. The influence of m on the classifier accuracy is here particularly evident. First, the CDT_ϵ achieves prompter detections at low values of m. Second, in case of large values of m, the reduction of the classification error after the change is slower even for both the proposed solution and the JIT based on CDT_X since less supervised samples are available to re-train the model after the detection of a change. The effect of classes' swap (*test ID 2*) is presented in Fig. 2. Such concept drift cannot be perceived by CDT_X and, hence, the JIT relying on the CDT_X provides the worst performance. Although the CDT_X can not perceive this concept drift (thus can not remove the obsolete samples from the classifier knowledge-base), the classification error decays thanks to the arrival of fresh supervised samples. Such decay varies depending on the amount of supervised samples provided: the larger is the amount of obsolete samples in the knowledge base of the classifier, the longer becomes the adaptation phase by simply introducing new supervised samples. These results corroborate the JIT approach in which, after detecting a concept drift, the classifier is always reconfigured by considering only the new (possibly recurrent) concept. In contrast, CDT_ϵ promptly detects the change and hence, both the JIT relying on CDT_ϵ and the proposed solution provide the best accuracy. The results of *test ID 3* (Fig. 3) show the peculiar situation in which classification problem becomes easier after the concept drift. In this case the CDT_X is prompter than CDT_ϵ, as such concept is more easy to perceive observing the distribution of the observations (at least when few supervised samples are available).

These three tests show the advantages provided by the joint monitoring of input observations and classification error: the proposed JIT adaptive classifier exploits the promptness in detecting changes provided by CDT_X (in particular when few supervised samples are available), and it is able to perceive changes that affect only the classification error (thanks to CDT_ϵ).

Test ID 4 and *5*, whose results are presented in Fig. 4 and Fig. 5, show the effectiveness of the proposed solution in case of recurring concepts. More in detail, the JIT able to exploit recurrent concepts guarantees the best performance after a transient concept drift (Fig. 4). It is worth noting that the advantages of handling recurrent concepts become evident when m increases: the plots show that at low supervised-sample rates the identification of recurrent concepts guarantees substantial improvements. Results in Fig. 5, which concern the sequence of abrupt concept drifts of *test ID 5*, show more clearly the advantages provided by the use of recurrent concepts; in particular, the classification error after the last change decreases more rapidly than the two previous changes, since both the realizations of concept C_1 (in the initial training

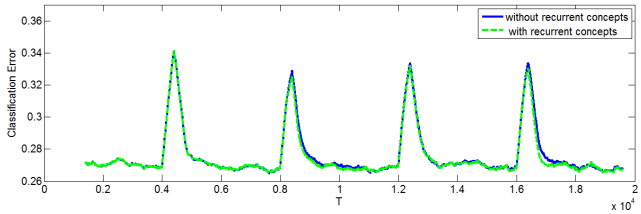


(a) $m = 2$

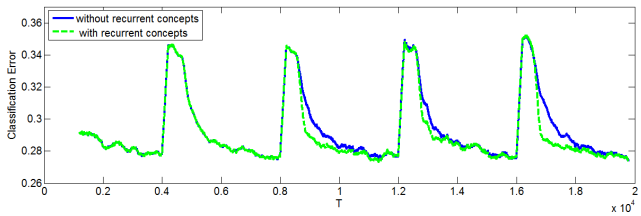


(b) $m = 20$

Fig. 4. Test ID 4: comparison of the classification errors as function of time for the proposed JIT adaptive classifier and the solutions based on CDT_X and on CDT_ϵ only.



(a) $m = 2$



(b) $m = 20$

Fig. 5. Test ID 5: comparison of the classification errors as function of time for the proposed JIT adaptive classifier and the solutions based on CDT_X and on CDT_ϵ only.

sequence and after the second concept drift) can be exploited.

VI. Conclusions

The paper presents a JIT adaptive classifier that is able to react to a concept drift resulting in either a change in the classification error or in a non-stationarity of the distribution of the unlabeled observations. As soon as a concept drift is detected, the new concept is associated with an automatically identified training sequence. Then, the previously encountered concepts are analyzed to determine if these are coherent with the current one and, in this case, the corresponding training sequences are jointly used to retrain the classifier. The proposed solution operates as an ensemble of classifiers framework, which are effectively able to handle recurrent concepts. We have shown that both the improved detection capabilities and the identification of recurrent concepts are essential elements for achieving satisfactory classification accuracy when the supervised information available during the operational life is scarce.

Acknowledgments

This research has been funded by the European Commission's 7th Framework Program, under grant Agreement INSFO-ICT-270428 (iSense).

References

[1] A. Tsymbal, "The problem of concept drift: definitions and related work. department of computer science, trinity college dublin," Ireland, Technical Report TCD-CS-2004-15,, Tech. Rep., April 2004.

[2] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence SBIA 2004*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3171, pp. 66–112.

[3] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[4] K. Nishida and K. Yamauchi, "Learning, detecting, understanding, and predicting concept changes," in *International Joint Conference on Neural Networks (IJCNN 2009)*. IEEE, 2009, pp. 2280–2287.

[5] C. Alippi and M. Roveri, "Just-In-Time adaptive classifiers – part I: Detecting nonstationary changes," *Neural Networks, IEEE Transactions on*, vol. 19, no. 7, pp. 1145 –1153, july 2008.

[6] ——, "Just-In-Time adaptive classifiers – part II: Designing the classifier," *Neural Networks, IEEE Transactions on*, vol. 19, no. 12, pp. 2053 –2064, dec. 2008.

[7] C. Alippi, G. Boracchi, and M. Roveri, "Adaptive classifiers with ICI-based adaptive knowledge base management," in *Artificial Neural Networks (ICANN 2010)*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6353, pp. 458–467.

[8] ——, "A Just-In-Time adaptive classification system based on the Intersection of Confidence Intervals rule," *Neural Networks*, vol. 24, no. 8, pp. 791 – 800, 2011.

[9] R. Elwell and R. Polikar, "Incremental learning in nonstationary environments with controlled forgetting," in *International Joint Conference on Neural Networks (IJCNN 2009)*. IEEE, 2009, pp. 771–778.

[10] S. Chen, H. He, K. Li, and S. Desai, "Musera: multiple selectively recursive approach towards imbalanced stream data mining," in *International Joint Conference on Neural Networks (IJCNN 2010)*. IEEE, 2010, pp. 1–8.

[11] G. Ditzler and R. Polikar, "An ensemble based incremental learning framework for concept drift and class imbalance," in *International Joint Conference on Neural Networks (IJCNN 2010)*. IEEE, 2010, pp. 1–8.

[12] R. Klinkenberg, "Learning drifting concepts: Example selection vs. example weighting," *Intelligent Data Analysis*, vol. 8, no. 3, pp. 281–300, 2004.

[13] C. Alippi, G. Boracchi, and M. Roveri, "Change detection tests using the ICI rule," in *International Joint Conference on Neural Networks (IJCNN 2010)*, 2010, pp. 1 –7.

[14] ——, "A hierarchical, nonparametric, sequential change-detection test," in *International Joint Conference on Neural Networks (IJCNN 2011)*, 31 2011-aug. 5 2011, pp. 2889 –2896.

[15] ——, "An effective Just-In-Time adaptive classifier for gradual concept drifts," in *International Joint Conference on Neural Networks (IJCNN 2011)*, 31 2011-aug. 5 2011, pp. 1675 –1682.

[16] ——, "Just in time classifiers: Managing the slow drift case," *International Joint Conference on Neural Networks (IJCNN 2009)*, vol. 0, pp. 114–120, 2009.

[17] G. S. Mudholkar and M. C. Trivedi, "A Gaussian approximation to the distribution of the sample variance for nonnormal populations," *Journal of the American Statistical Association*, vol. 76, no. 374, pp. pp. 479–485, 1981.

[18] A. Goldenshluger and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression," *Math. Meth. Statistics*, vol. 6, pp. 135–170, 1997.

[19] R. Johnson and D. Wichern, *Applied multivariate statistical analysis*. Prentice Hall, 2002, no. v. 1.