Just-in-Time Adaptive Classifiers—Part II: Designing the Classifier

Cesare Alippi, *Fellow, IEEE*, and Manuel Roveri

***Abstract*—Aging effects, environmental changes, thermal drifts, and soft and hard faults affect physical systems by changing their nature and behavior over time. To cope with a process evolution adaptive solutions must be envisaged to track its dynamics; in this direction, adaptive classifiers are generally designed by assuming the stationary hypothesis for the process generating the data with very few results addressing nonstationary environments. This paper proposes a methodology based on k-nearest neighbor (NN) classifiers for designing adaptive classification systems able to react to changing conditions just-in-time (JIT), i.e., exactly when it is needed. k-NN classifiers have been selected for their computational-free training phase, the possibility to easily estimate the model complexity k and keep under control the computational complexity of the classifier through suitable data reduction mechanisms. A JIT classifier requires a temporal detection of a (possible) process deviation (aspect tackled in a companion paper) followed by an adaptive management of the knowledge base (KB) of the classifier to cope with the process change. The novelty of the proposed approach resides in the general framework supporting the real-time update of the KB of the classification system in response to novel information coming from the process both in stationary conditions (accuracy improvement) and in nonstationary ones (process tracking) and in providing a suitable estimate of k. It is shown that the classification system grants consistency once the change targets the process generating the data in a new stationary state, as it is the case in many real applications.**

***Index Terms*—Intelligent systems, learning systems, neural networks, pattern classification.**

I. Introduction

In world applications, processes are characterized by an evolutionary nature and may change their behavior over lifetime. In general, this is due to ageing effects, thermal drifts, nonstationary phenomena, soft and hard faults that may affect industrial, environmental and natural processes and, as a consequence, data coming from them. At the same time, when no change occurs, the process might provide additional information that could be exploited to improve the application accuracy. The former case is common in all applications envisaging measurement systems and sensors (the dynamic of the change generally depending on the sensor nature and its interaction with the environment). For instance, a change in stationarity is generally particularly evident in data coming from X-ray detectors, electronic noses, monitoring systems for aquatic, mountain and desertic environments, and, more in general, in all those measurement systems operating in harsh environments, while it is somehow milder (i.e., ruled by slow dynamics) when the deployment refers to noncritical environments. The latter aspect, i.e., integration of fresh information to improve the accuracy of the classification system, requires the presence of a supervisor. In quality analysis applications, a supervisor is present that randomly selects artifacts or goods from industrial/manufacturing processes and assesses their quality through a direct inspection, identifies false positive or negative samples in electronic olfactory systems, or corrects items in a character/word recognition ones; such supervised information, provided it is suitably integrated in the classifier, leads to accuracy improvements.

In solutions developed for all above applications, the process evolution requires adaptive mechanisms to optimally exploit information coming from the process. Solutions, whatever they have been designed for, must hence be reactive to provide, through adaptation, the best possible performance.

Traditional adaptive classifiers [1]–[6] assume the stationarity hypothesis for the process generating the data; as such, they provide good results in stationary conditions and are scarcely effective in nonstationary environments.

There exists a limited literature addressing adaptive classifiers in nonstationary conditions, with research focusing on adaptive preprocessing techniques, adaptive neural networks, and adaptive classifiers for specific applications. Transformations invariant to environmental changes have been suggested in [7]–[9]. Li *et al.* [10] propose a classification design with adaptive filtering mechanisms based on *a priori* information of the process. A time adaptive self-organizing map (SOM), automatically adjusting the network parameters to work both in stationary and nonstationary environments, is presented in [11]; Carpenter *et al.* [12] propose a fuzzy ARTMAP as a nonparametric probability estimator for nonstationary pattern recognition problems and [13] a probabilistic neural network for classifying patterns characterized by time-varying distributions. In [14], an online tracking of analog neurons subject to slowly drifting weights is proposed, while Kuh [15] provides a comparison of tracking algorithms for single-layer threshold networks in presence of random drifts.

Few papers consider adaptive classifiers in applications, e.g., control in manufacturing systems [16], neural networks for breast cancer detection [17], adaptive image analysis for aerial surveillance [18], condition monitoring and fault prediction [19], visual surveillance of noncooperative and camouflaged targets [20], and induction machine stator fault diagnostics [21].

The main limit of solutions proposed in the related literature is the need of continuously updating the knowledge base (KB) of

Manuscript received March 15, 2007; revised February 05, 2008; accepted April 29, 2008. Current version published November 28, 2008.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy (e-mail: alippi@elet.polimi.it; roveri@elet.polimi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2003998

the classifier (i.e., training set) or its network weights even when unnecessary (no stationarity tests are carried out). Moreover, most of these approaches neither adequately describe the effects of the introduction/removal of fresh information into an existing KB nor address the relationships between what proposed and the asymptotic behavior defined by the theory in the stationary case.

Here, we suggest a design methodology for adaptive classifiers which, by exploiting supervised information coming from the field during operational life, modifies, whenever appropriate, the KB characterizing the classifier to maintain/improve classification accuracy. In particular, the classifier *exploits fresh available information in stationary conditions* and *reacts to the changing environment in nonstationary* ones to track the system change. The novelty of the approach resides in the possibility to update just-in-time (JIT), i.e., exactly when needed or convenient, the classifier and, at the same time, provide a fully automatic adaptive framework. Designing a JIT adaptive classifier involves the following two steps:

1) definition of methods able to identify the most convenient instant of time for integrating the incoming fresh knowledge (data) in the classifier (issue addressed in [22]);
2) integration of such information in the KB of the classifier to improve accuracy/track the process change (issue addressed in this paper).

The joint use of nonstationary detection tests and adaptive knowledge mechanisms allows us for designing JIT adaptive classifiers that improve their accuracy (whenever new information is available) in stationary conditions (so as to follow the asymptotic learning theory) and react to changes in nonstationary conditions by tracking the evolution of the data generating process. It is shown that the JIT adaptive classifier, based on k-nearest neighbor (NN) classification systems for their training free operational modality and easiness in manipulating model and computational complexity, is consistent in stationary conditions or when the process gains a stationary state. In other scenarios, the JIT adaptive classifier keeps tracking the process evolution to reduce the classification error.

The structure of this paper is as follows. Section II introduces the JIT classifier addressing stationary conditions, its relationship with the asymptotic theory, and how to estimate the needed parameters. The general framework design for JIT adaptive classifiers dealing with both stationary and nonstationary situations is presented in Section III. Experimental results are provided in Section IV.

II. JIT Adaptive Classifiers in Stationary Conditions

The operational framework onto which JIT classifiers for stationary conditions operate is that of asymptotical learning [21]. There, *consistency* guarantees that the performance of consistent classifiers asymptotically converges to the optimal Bayes' one [23] when the training set (or KB) composed of n independent and identically distributed (i.i.d.) pairs increases. This means that, in consistent learning methods, it is always worth integrating fresh samples in the classifier to improve accuracy (the compromise between cost of integration and accuracy gain can be studied with sequential analysis methods [24]).

Among different classification families, k-NNs [25] are particularly appealing consistent classifiers to be used as a core for JIT classification systems due to the absence of a proper training phase and an easy management of the classifier complexity. In fact, given n and estimated k, the k-NN-based JIT can be made adaptive and automatic through a suitable management of its KB.

The asymptotic consistency of k-NN classifiers is granted in the stationary case provided that i.i.d. training samples are extracted from the process generating the data and that k grows less than linearly with n [26], [27]. The optimal value of k as n increases, namely, the value of k to be used in the JIT classifier to minimize the generalization error given n training samples, must be estimated (bad estimates would reflect on classification performance) and represents a key point in constructing an automatic adaptive k-NN-based classifier.

In this direction, Fukunaga [28] has shown that the optimal k given n samples for the k-NN density estimate is

$$k_{\mathrm{FUK}}(n) = \left[\frac{d(d+2)^2\pi^2 \int p^2(x)dx}{\Gamma^{4/d}\left(\frac{d+2}{2}\right)\int \alpha(x)^2 p^{2-4/d}(x)dx}\right]^{\frac{d}{d+4}} n^{\frac{4}{d+4}} \tag{1}$$

where Γ is the gamma function, $p(x)$ is the probability density function of the input variable x, d is the size of the input space, and $\alpha(x)$ is a function resulting from a second-order Taylor expansion of $p(x)$. Moreover, when n increases, $C(k_{\mathrm{FUK}}(n))$ asymptotically tends to the Bayes classifier [23], [29].

Unfortunately, determination of the optimal k requires *a priori* information such as the $p(x)$, which is hardly available in real applications.

A different approach, which does not require unrealistic hypotheses, suggests k to be estimated with leave-one-out (LOO) [30]. Unfortunately, a LOO approach is computationally expensive, in particular, when n gets large, and cannot be considered every time a new sample (or a limited set of samples of cardinality Δn) is available. A viable solution to this problem is to integrate LOO and Fukunaga's approaches to provide a low computationally intensive estimate of k as n increases. In particular, by taking the ratio $k_{\mathrm{FUK}}(n+\Delta n)/k_{\mathrm{FUK}}(n)$, we obtain a recursive estimate for k

$$k_{\mathrm{FUK}}(n+\Delta n) = k_{\mathrm{FUK}}(n)\left(\frac{n+\Delta n}{n}\right)^{\frac{4}{d+4}}. \tag{2}$$

When we are not in possess of $p(x)$, unknown $k_{\mathrm{FUK}}(n)$ can be evaluated with $k_{\mathrm{LOO}}(n)$. As a consequence, the k to be considered having collected Δn additional training samples from n is

$$k(n+\Delta n) = k_{\mathrm{LOO}}(n)\left(\frac{n+\Delta n}{n}\right)^{\frac{4}{d+4}}. \tag{3}$$

Once estimated $k_{\mathrm{LOO}}(n)$, the new $k(n+\Delta n)$ can be directly computed with (3), which proves to be effective in the neighborhood of n. In particular, from experiments, (3) provides a good estimate $k(n+\Delta n)$ for all values $\Delta n = \{1, 2, \ldots, \Delta n_{\max}\}$ with $\Delta n_{\max}$ accounting for about 25% of n. Once we have received and integrated in the classifier (either one by one or through small sets of samples) more than $\Delta n_{\max}$ samples, a new $k_{\mathrm{LOO}}(n)$ should be evaluated over all received $n+\Delta n_{\max}$ data.

More in detail, the classifier given in Algorithm 1 operates as follows.

Algorithm 1: Adaptive Classifier (x)

```
1. Estimate $k_{\text{LOO}}$ on the initial knowledge base $\text{KB}_0$ and set $n = |\text{KB}_0|$;
2. $k(n) = k_{\text{LOO}}$;
3. $\text{KB} = \text{KB}_0$; $n_{\text{IKB}} = 0$; $\Delta n = 0$;
4. while (1) {
5.   if (new information IKB is available) {
6.     $\text{KB} = \text{KB} \cup \text{IKB}$;
7.     $n_{\text{IKB}} = |\text{IKB}|$; $\Delta n = \Delta n + n_{\text{IKB}}$
8.     if $\Delta n/n \geq 0.25$ {
9.       Estimate $k_{\text{LOO}}$ on KB and set $n = |\text{KB}|$;
10.        $n_{\text{IKB}} = 0$; $\Delta n = 0$;
11.      }
12.    $k(n + \Delta n) = k_{\text{LOO}}(n)((n + \Delta n)/n)^{4/(d+4)}$;
13.  }
14.  if ($x$ is available) {
15.    classification = $k$-NN$(x, \text{KB}, k(n + \Delta n))$;
     }
16. }
```

The initial knowledge base KB_0 characterizing the initial k-NN classifier is used to estimate the initial value $k(n)$ by means of a LOO estimate (step 1). When new knowledge IKB is provided to the classifier (step 5), it is inserted in the KB of the classifier (step 6) and the new value of $k(n)$ is computed (step 12). When ratio $\Delta n/n$ between the number of new samples and the number of samples in the KB is above 0.25 (step 8), a new value of $k(n)$ is estimated with LOO applied to $n + \Delta n$ samples (step 9). Obviously, the designer can opt for a different strategy and set a threshold for activating the KB updating procedure based on *a priori* information/computational constraints. Finally, the new incoming input sample x is classified with the k-NN rule based on KB and $k(n + \Delta n)$.

An increment in the cardinality of the training base is always profitable from the accuracy point of view but it also induces a computational complexity which grows as $O(kn)$ (and could become critical in embedded solutions characterized by contained computational and memory resources). As such, even if the whole KB must be kept for information management, in stationary conditions, it is generally profitable to reduce the cardinality of the KB of the k-NN from KB to $\text{KB}_{\text{reduced}}$ to speedup the computation and reduce memory consumption during the operational life of the classifier (i.e., step 15 of algorithm 1 should be modified as "classification $= k$-NN$(x, \text{KB}_{\text{reduced}}, k(n + \Delta n))$." At the same time, data reduction techniques support outliers removal.

The aim of condensing and editing techniques is to maintain the smallest set of training instances yet preserving the classification accuracy on the training set. This can be achieved by removing both possible outliers (noisy samples that cause wrong classification) and superfluous samples (which are redundant, in the sense that they are dominated by others). The condensed nearest neighbor (CNN) rule [33] and the reduced nearest neighbor (RNN) [34] one move in this direction. However, such methods are advisable only if the Bayes error is small, otherwise the effectiveness in reduction might be poor.

Differently, Wilson's editing rule [35] removes an instance from the training set whenever it does not agree with the majority of its k neighbors (typically $k = 3$). Noisy instances (outliers), as well as borderline samples, are removed from the training set, hence providing a smoother decision boundary.

There it also exists a particularly elegant geometrical solution which aims at reducing the KB in k-NN classifiers and, at the same time, preserving the decision boundary [36]. The method requires generation of a Voronoi diagram in a highly dimensional space, operation proven to be computationally expensive. Hence, for practical applications, we suggest to consider approximate solutions such as the Gabriel graph rule [36] and the relative neighbor graph rule [36] that solve the problem by trading off computational complexity and performance.

III. JIT Adaptive Classifiers in Nonstationary Environments

When the process under investigation changes its statistical behavior during operational life, which means that at a certain instant of time the probability density function (pdf) underlying the process changes, the training set contains obsolete information which negatively impacts classification performance. The main consequence of nonstationarity is hence a loss in accuracy, which can be contained by considering adaptive classification systems designed to track the process (exploitation of fresh knowledge and removal of obsolete one).

The joint use of change detection tests and dynamic knowledge management leads to the JIT adaptive classifier given in Algorithm 2 operating both in stationary and nonstationary conditions. The classifier, at this high abstraction level, is optimal and works as follows.

Algorithm 2: General JIT Adaptive Classifier (x)

```
1. Configure the classifier on the initial knowledge base $\text{KB}_0$;
2. Configure the nonstationarity detection test on $\text{KB}_0$;
3. $\text{KB}(0) = \text{KB}_0$;
4. $i = 1$;
5. while (1) {
6.   if (new knowledge $\text{IKB}(i)$ is available) {
7.     $\text{KB}(i) = \text{Add}(\text{KB}(i-1), \text{IKB}(i))$;
8.     Reconfigure the classifier on $\text{KB}(i)$;
9.     $i = i + 1$; }
10.  if (nonstationarity detection test $(x)$ = Stationary)
11.    Use the classifier on the input sample $(x)$ as per algorithm 1;
12.  else {
13.    Identify the obsolete knowledge $\text{KB}_{\text{OBS}}$ in the knowledge base $\text{KB}(i)$;
14.    $\text{KB}(0) = \text{Rem}(\text{KB}(i), \text{KB}_{\text{OBS}})$;
15.    Reconfigure the classifier on $\text{KB}(0)$;
16.    Reconfigure the nonstationarity detection test on $\text{KB}(0)$;
17.     Use the classifier on the input sample ($x$) relying on KB(0);
18.     $i = 1$;
19.     }
20. }
```

The initial knowledge base KB_0 is used to configure both the initial classifier (step 1) and the nonstationary change detection test (step 2). When new information (IKB) is provided to the classifier (step 6), the algorithm suitably integrates it in the KB (step 7) and reconfigures the classifier accordingly (step 8). If the process under monitoring remains stationary and new knowledge is not available during operational life, the methodology works as per stationary environments (step 11) following algorithm 1. Conversely, when the change detection test detects a nonstationary behavior for the process (step 12), obsolete information needs to be removed from the KB (step 14), a reconfiguration of the classifier (step 15) and the nonstationary change detection test are invoked (step 16). The new process becomes the reference one for detecting subsequent evolutions. In this case, the classification of the input sample x is performed on the updated KB after the reconfiguration of the classifier (step 17).

The general methodology for developing JIT adaptive classifiers of Algorithm 2 provides a comprehensive solution to the classification problem in stationary/nonstationary environments: new information is inserted in the classifier in stationarity conditions to improve accuracy and tracking the process evolution is performed in the nonstationary case. However, to effectively design a JIT adaptive classifier, designers have to take several implementation choices regarding the classifier (structure, family, and model), the change detection test to be used, the data onto which apply the change detection test, and the updating mechanism for the KB.

Again, we suggest to use k-NN classifiers for their immediate training phase and easy knowledge management. As shown in Appendix I, the choice is supported from the theory: a JIT classifiers based on the k-NN rule achieves the Bayes classifier in the stationary case and whenever the nonstationarity moves the process in a new stationary state (consistency in probability). In other scenarios (e.g., continuous drift), general consistency cannot be guaranteed depending on the dynamic of the change; yet, the JIT adaptive classifier tracks the process evolution to reduce the classification error.

As far as the change detection issue is concern, there exists a large literature which generally requires *a priori* information about the process generating the data to properly configure the test parameters. We suggest to use the adaptive self-configuring statistical CI-CUSUM test suggested by the authors in the companion paper [22] to detect nonstationarity trends, which combines effective change detection abilities in *a priori* information-free context with a contained computational complexity burden. However, the designer can select his/her favorite change detection method.

A last issue to be faced refers to obsolete information removal and insertion of new supervised knowledge. Removal of obsolete data from the KB, which is performed in step 14, can be addressed with solutions based on oblivion coefficients [37], outlier detections [35], [38], and sample regularization based on the information content [35]; here, we assume that the last acquired samples are the most meaningful ones and, as such, represent the current state of the process. The direct consequence is that older samples are obsolete and can be removed from the KB. The assumption is acceptable in almost all real situations where nonstationarity is associated with process aging and faults.

The minimum number of samples T to be kept in the KB after a removal operation should allow both for granting accuracy and satisfying the operational requirements of the change detection test (for instance, the CI-CUSUM change detection test requires a minimum of $T > 500$ samples). Aspects related to the number of samples to be considered and their relationship with the dynamic of process under monitoring are provided in Section IV-B.

The JIT adaptive classifier of Algorithm 2 tailored to k-NN and the CI-CUSUM test is finally presented in Algorithm 3.

Algorithm 3: JIT Adaptive Classifiers (x)

```
1. Estimate $k_{\mathrm{LOO}}$ on $\mathrm{KB}_0$ and set $k(n) = k_{\mathrm{LOO}}$, $n = |\mathrm{KB}_0|$;
2. Configure the classifier on $\mathrm{KB}_0$;
3. Configure the nonstationarity detection test on $\mathrm{KB}_0$;
4. $\mathrm{KB} = \mathrm{KB}_0$; $n_{\mathrm{IKB}} = 0$; $\Delta n = 0$;
5. **while** (1) {
6.   **if** (new knowledge IKB is available) {
7.       $\mathrm{KB} = \mathrm{KB} \cup \mathrm{IKB}$;
8.       $n_{\mathrm{IKB}} = |\mathrm{IKB}|$; $\Delta n = \Delta n + n_{\mathrm{IKB}}$;
9.       if $\Delta n / n \geq 0.25$ {
10.          Estimate $k_{\mathrm{LOO}}$ on KB and set $n = |\mathrm{KB}|$;
11.          $n_{\mathrm{IKB}} = 0$; $\Delta n = 0$;
12.      }
13.      $k(n + \Delta n) = k_{\mathrm{LOO}}(n)((n + \Delta n)/n)^{4/(d+4)}$;
14.  }
15.  **if** (CI-CUSUM test (sample $x$) = Stationary)
16.      classification = $k$-NN$(x, \mathrm{KB}, k(n + \Delta n))$;
17.  **else** {
18.      $\mathrm{KB}_{\mathrm{OBS}}$ = old knowledge (more than last $T$ samples)
19.      $\mathrm{KB} = \mathrm{Rem}(\mathrm{KB}, \mathrm{KB}_{\mathrm{OBS}})$;
20.      $n = |\mathrm{KB}|$; $n_{\mathrm{IKB}} = 0$; $\Delta n = 0$;
21.      Estimate $k_{\mathrm{LOO}}$ on KB and set $k(n) = k_{\mathrm{LOO}}$;
22.      Configure the classifier on KB;
23.      Configure the nonstationarity detection test on KB;
24.      classification = $k$-NN$(x, \mathrm{KB}, k(n))$;
25.  }
26. }
```

A Critical View of the Suggested Adaptive Algorithm: The proposed methodology relies on the ability to detect changes in the process under investigation and adapt the classifier to track the process.

When the change detection test does not detect changes when there they are (false negative), the JIT adaptive classifier cannot commute into the nonstationary operational mode (step 12 of

Algorithm 2) and it behaves as a traditional stationary classifier that gains new knowledge during operational life.

The algorithm performance might not be satisfactory when the "last samples characterize the process under investigation" assumption does not hold. However, this hypothesis generally holds in the practice provided that the dynamic of the change is slower than the tracking ability of the JIT adaptive classifier (e.g., the process changes so quickly that most of the last T samples are not representative of the current state of the process). Note that we are not requiring the process changes to be slow (that could be an unacceptable assumption) but that such variations are slower (e.g., ten times) than the reaction ability of the JIT adaptive classifier. This last hypothesis generally holds if we consider abrupt and aging change models proposed in the literature [39], [40].

Unfortunately, when the dynamic of the process under investigation is faster than the tracking ability of the JIT classifier, last acquired samples are already obsolete and the JIT classifiers may lead to unsatisfactory results (in feedforward neural-network-based classifiers, we will update weights online, at each instant of time, to cope with such fast evolution, as suggested in [15]).

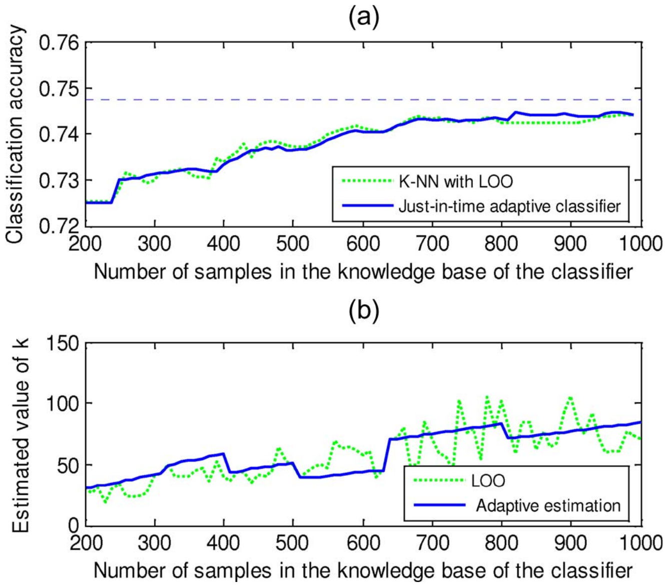

Fig. 1. Application D1. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN; dashed line: Bayes classifier). (b) k estimated during the experiment (solid line: k evaluated according to (4); dotted line: k_{LOO}).

IV. Experimental Section

Applications considered in the companion paper are here envisaged to give continuity to the exposition. Applications D1 and D3 are simulated while D2 and D5 refer to real experiments; application D4 is missing because it cannot be applied. More in detail, we have the following.

- *Application D1* refers to a synthetic monodimensional classification problem with two equiprobable classes (ω_0, ω_1) each of which ruled by a Gaussian distribution $p(x|\omega_0) = N(0,3)$ and $p(x|\omega_1) = N(4,3)$.
- *Application D2* refers to the SATIMAGE benchmark [41] and specifically addresses Landsat multispectral scanner images (each sample is characterized by 36 features) aiming at classifying the nature of the soil.
- *Application D3* refers to gas-self-assembled-monolayers (SAM) [42], [43] sensors. The application considers a set of five SAM gas sensors, two features (the sensor measurement and its derivative) extracted from each signal.
- *Data Set D5* refers to the physiological data benchmark suggested in [44]. The data set contains two measurements retrieved from naps of healthy people: the respiratory signal (RPS) and the electrooculography signal (EOG).

The experimental campaign is organized into two subsequent steps. We evaluate and discuss first the behavior and performance of the proposed JIT adaptive classifier working in the stationary case and then in the nonstationary one.

A. *JIT Adaptive Classifiers in Stationary Conditions*

Three performance indexes have been considered:

- classification accuracy evaluated with cross validation;
- k estimated according to (4);
- computational time (CT) defined as the execution time (in seconds) needed to perform classification (reference platform: Intel Centrino 1.7 GHz, 1-GB RAM, Windows XP; unnecessary processes aborted).

To compare performance of the k-NN with k evaluated according to (4), we considered a reference traditional k-NN classifier operating on the same KB. In the reference k-NN classifier, the optimal k is estimated with a LOO procedure (invoked in correspondence of each new sample, hence constituting the best operational condition).

Application D1: Fig. 1(a) shows the classification accuracy of the JIT and the reference classifier as function of the number of samples in the KB n while Fig. 1(b) presents k evaluated according to (4) and the leave-one-out k_{LOO}. When n increases, the classification accuracy of the JIT classifier increases as well (in line with the expected asymptotical behavior) and tends to the optimal Bayes value (horizontal dashed line). Obviously, the same holds for the reference k-NN classifier based on k_{LOO} but at cost of a higher computational time (see also Table I).

By inspecting Fig. 1(b), we note that an increment in n implies an expected increment in k and that k_{LOO} suffers from high variance as shown by the theory, e.g., see [30] and [31]. Even if such variability also influences the initial value $k_{\mathrm{LOO}}(n)$ to be used in (4), subsequent $k(n+\Delta n)$ values are regular until a new value $k_{\mathrm{LOO}}(n)$ needs to be estimated. The sudden changes in Fig. 1(b) are associated with a new value inserted in (4), according to step 9 in Algorithm 1.

Application D2: Comments provided for application D1 are valid also for application D2. From Fig. 2, we appreciate the agreement in accuracy and k between the JIT classifier and the reference one.

We observe that, according to (4), $k(n)$ grows at a slower rate with respect to (w.r.t.) n in D2 than in D1; this phenomenon is generated by size of the input space which is much larger in D2 $(d = 36)$ than in D1 $(d = 1)$: the larger the d, the smaller the required increment in k, as it can be immediately seen from (4) where d is at the exponent. Again, the computational complexity is in favor of the JIT classifier (see Table I).

TABLE I
SIMULATION RESULTS OF THE JIT ADAPTIVE CLASSIFIER AND THE REFERENCE k-NN IN STATIONARY CONDITIONS

		k-NN with LOO			JIT Adaptive Classifier			$E\left[\frac{k_{LOO}(n)-k(n)}{k_{LOO}(n)}\right]$	
		Initial Accuracy (%)	Final Accuracy (%)	CT(s)	Initial Accuracy (%)	Final Accuracy (%)	CT(s)	Mean	Var.
D1		73.01	74.20	126.1	73.0	74.31	6.2	0.20	0.06
D2		83.7	87.6	4779.9	83.7	87.6	140.1	0.38	0.79
D3		93.8	96.6	1356.2	93.6	96.6	23.8	0.18	0.15
D5	EOG	59.6	60.1	1831.1	59.6	60.1	44.2	0.26	0.15
	RPS	55.5	56.5	1920.4	55.4	56.5	43.1	0.29	0.23

TABLE II
NONSTATIONARY CHANGES FOR APPLICATION D1

Original process	$D1(t)=\begin{cases}p(x\mid\omega_0)=N(100,3)\\ p(x\mid\omega_1)=N(104,3)\end{cases},\quad 1\ge t\ge 5000$
Abrupt	$D1(t)=\begin{cases}p(x\mid\omega_0)=N(105,3)\\ p(x\mid\omega_1)=N(109,3)\end{cases},\quad 5001\ge t\ge 10000$
Slow change of state	$D1(t)=\begin{cases}p(x\mid\omega_0)=N(100+5\frac{t-1000}{1000},3)\\ p(x\mid\omega_1)=N(104+5\frac{t-1000}{1000},3)\end{cases},\quad 5001\ge t\ge 6000$ $D1(t)=\begin{cases}p(x\mid\omega_0)=N(105,3)\\ p(x\mid\omega_1)=N(109,3)\end{cases},\quad t>6001$

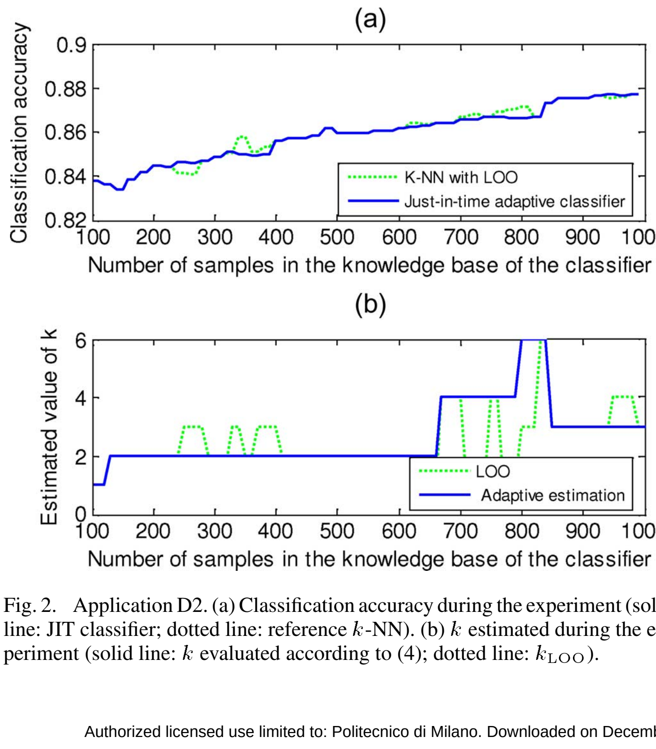

Fig. 2. Application D2. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN). (b) k estimated during the experiment (solid line: k evaluated according to (4); dotted line: k_{LOO}).

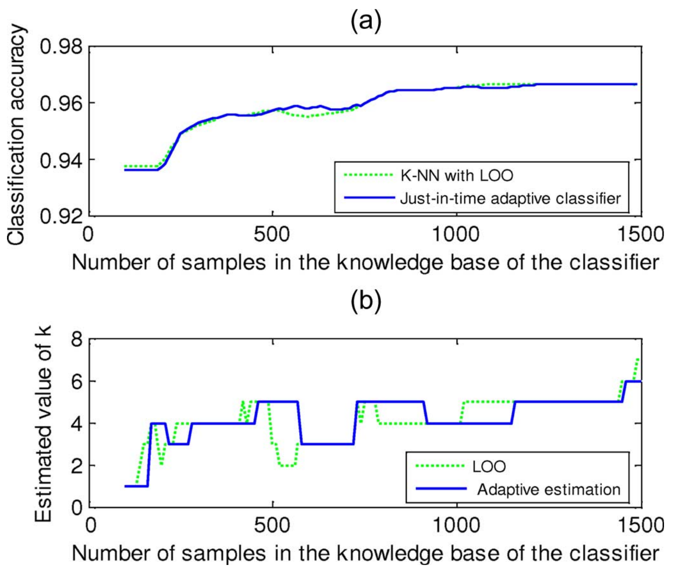

Fig. 3. Application D3. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: k-NN with LOO). (b) Estimated k during the experiment (solid line: k evaluated according to (4); dotted line: k_{LOO}).

Application D3: Results are given in Fig. 3 and comments are in line with that of application D2. Even in this case, the $k(n)$ function grows smoothly w.r.t. n $(d=10)$.

Application D5: Fig. 4.1(a) and 4.2(a) shows the classification accuracy of the JIT adaptive classifier and the reference k-NN for the respiratory signal measurements (NRPS) and the electrooculography signal measurements (NPOG), respectively. As depicted, the proposed algorithm guarantees a classification accuracy that is comparable with the one provided by the reference k-NN and, at the same time, it significantly reduces the required computational time (see Table I).

Again, the sudden variations in the k estimated according to (4) are associated with the need of periodically estimating a new k_{LOO} value.

Quantitative results have been summarized in Table I. In particular, the table shows the accuracy at the beginning and the

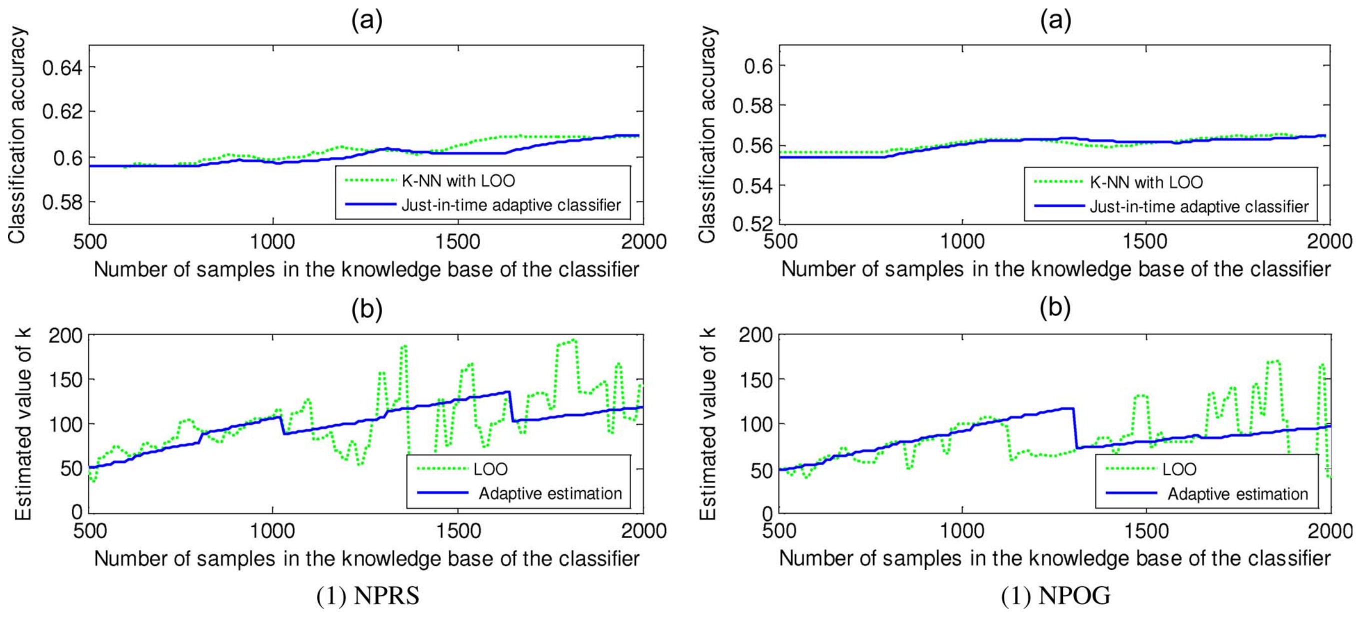

Fig. 4. Application D5: NPRS and NPOG. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: k-NN with LOO). (b) Estimated k during the experiment (solid line: k evaluated according to (4); dotted line: k_{LOO}).

end of the experiments for the JIT adaptive classifiers and the reference k-NN (asymptotic behavior).

The inclusion of new knowledge during operational life allows both approaches for improving their accuracy over time. The fact that the JIT classifier performance is in line with the optimal reference one is appreciable but the former significantly reduces the computational time due to the computationally cheaper estimate of k in the JIT classifier provided by (4).

B. JIT Adaptive Classifiers in Stationary/Nonstationary Conditions

Consider as performance indexes the following:

- accuracy as defined in the stationary case;
- the cardinality n of the KB at time t.

A nonadaptive traditional k-NN classifier (which relies only on the initial KB) was considered as a reference.

Application D1: Since application D1 is synthetic, we piloted it by considering both abrupt and drift changes affecting the process (see Table II for the characterization of the change nature).

Fig. 5.1(b) shows the average classification accuracy of the JIT adaptive classifier during the experiment with respect to the reference k-NN one. We observe that, while the k-NN permanently suffers from the presence of a change (the accuracy drops after the change), the JIT adaptive classifier reacts to it. In particular, the JIT accuracy decreases temporarily until the CI-CUSUM test detects the change in the data generating process (around $t = 6000$), the knowledge management procedure of Algorithm 3 is then activated, obsolete knowledge is removed from KB according to the adaptive T mechanism of Appendix B, and new information is added to the KB. Since the data generating process commutes from a stationary condition into a new stationary one after the change, the accuracy of the JIT classification system asymptotically converges to the performance of the Bayes classifier (as demonstrated in Section IV-A).

The reaction time mainly depends on the strength of the change and the sensibility of the change detection test. The tracking phenomenon can be studied in detail by inspecting Fig. 5.1(b) and 5.1(c). From 5.1(b), the classifiers experiences a loss in classification accuracy immediately after the abrupt change, which arises at time 5000.

At the beginning, just after the change, only few samples are representative of the new state and moderately affect accuracy. As data coming from the new state arrive and are integrated in the KB, their impact on performance increases (and classification accuracy decreases). Finally, the change is detected and no more samples are simply inserted into the KB (the knowledge management mechanisms of Algorithm 2 is activated). The result is that fresh data related to the new process state (and constituting data set Z_n) are inserted in the KB and obsolete ones (and constituting data set Z_{old}) removed: once Z_n dominates Z_{old}, accuracy starts increasing again and reaches the new steady state (detailed relationships between Z_n and Z_{old} are given in Appendix I). Finally, once the drift has exhausted its transient phase, the JIT classifier commutes back to the stationary mode.

Of course, no chances are given to the reference classifier which, being not able to modify itself and tracking the new stationary state, simply integrates new samples in its KB. The result is that it introduces a bias error associated with the presence of Z_{old} (see Appendix I).

As presented in Fig. 5.2(a), the situation is similar in correspondence of a slow drift targeting the process into a new stationary state.

Application D2: Results associated with this real application are given in Fig. 6.

From Fig. 6(b), we discover that the process is stationary and no changes are detected. In fact, n increases linearly without any intervention of the KB management mechanism. Fig. 6(a)

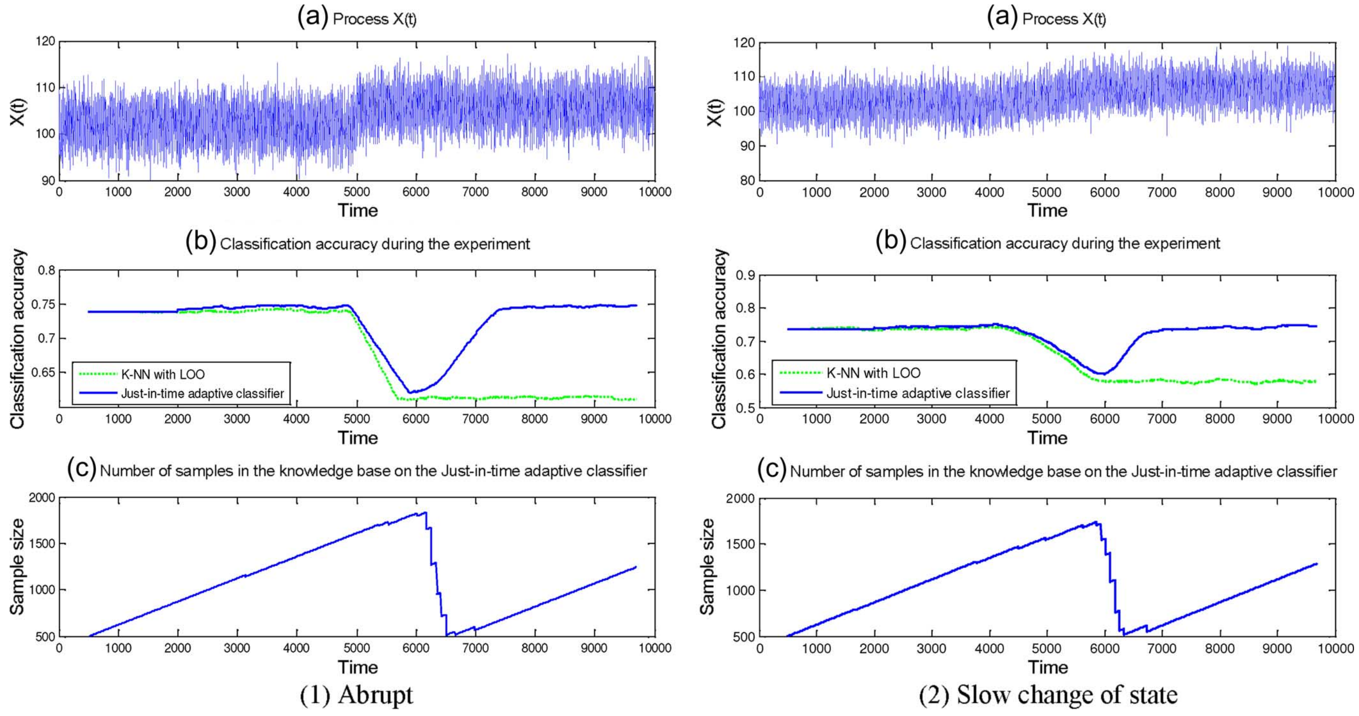

Fig. 5. Application D1. (a) Value of the process during the experiment. (b) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN). (c) Number of samples in the KB of the JIT adaptive classifier during the experiment.

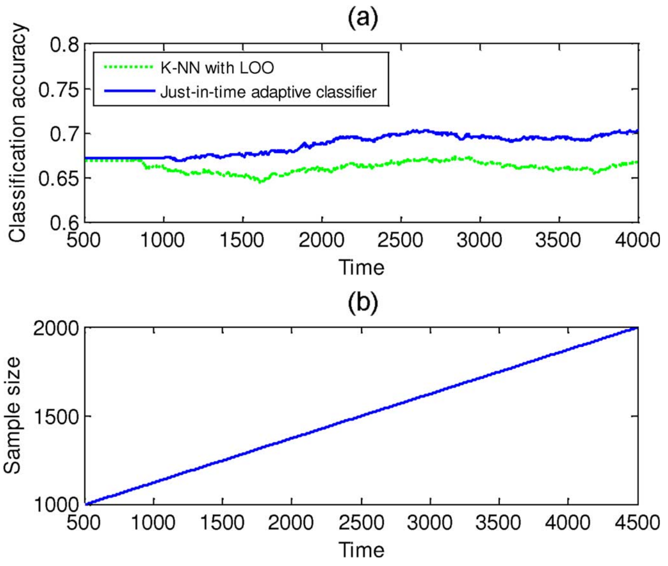

Fig. 6. Application D2. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN). (b) Number of samples in the KB of the JIT classifier during the experiment.

shows that the classification accuracy of the JIT classifier is in line with that of the reference k-NN one; the slight increment in performance of the former is due to the asymptotic behavior of the JIT classifier (which increases its accuracy as the number of samples in the KB increases) whereas the reference k-NN is not, being configured on KB_0.

Application D3: We applied, to application D3, an abrupt change according to a multiplicative perturbation model in which the generic parameter P is affected by a perturbation δ modifying its value from P^0 to $P^1 = P^0(1+\delta)$ and a drift change where parameter P^0 linearly evolves so that, at the end of the experiment, it assumes value $P^1 = P^0(1+\delta)$. Perturbation δ is uniformly extracted in each experiment from the $U(-2,2)$ interval and affects the resistive parameter of each SEM sensor.

As presented in Fig. 7.1(a), we observe that the classification accuracy of the reference k-NN classifier drops after the change $(t = 5000)$ and it remains there. Differently, the JIT classifier suffers only temporarily from the change and then reacts accordingly as in application D1. The size of the KB in the JIT adaptive classifier increases up to the change detection (around $t = 5500$) where the knowledge management mechanisms are activated. It is interesting to observe that the change detection test identifies changes even after $t = 5500$ (little saw-like shapes modulated on the main linear segments), which must be intended as change detections associated with small perturbations; see [22].

Afterwards, since no other changes are detected in the experiment (the process after the change achieves a new stationary state), the size of the KB continues increasing hence exploiting incoming fresh information.

The difference between the accuracy of the JIT adaptive classifier and the reference k-NN is relevant also in the drift change (Fig. 7.2). The JIT adaptive classifier reacts to the change and remains tracked to process while the reference k-NN classifier heavily suffers from the presence of a continuous drift.

As a last note, we observe a saw-like behavior of n in Fig. 7.2(b). Such behavior is due to the continuous drift: the change detection test proceeds identifying changes in stationarity hence activating the adaptive mechanism of the JIT.

Application D5: As presented in Fig. 8.1(a) and 8.2(a), the difference between the classification accuracy of the JIT adaptive classifier and the reference k-NN increases progressively

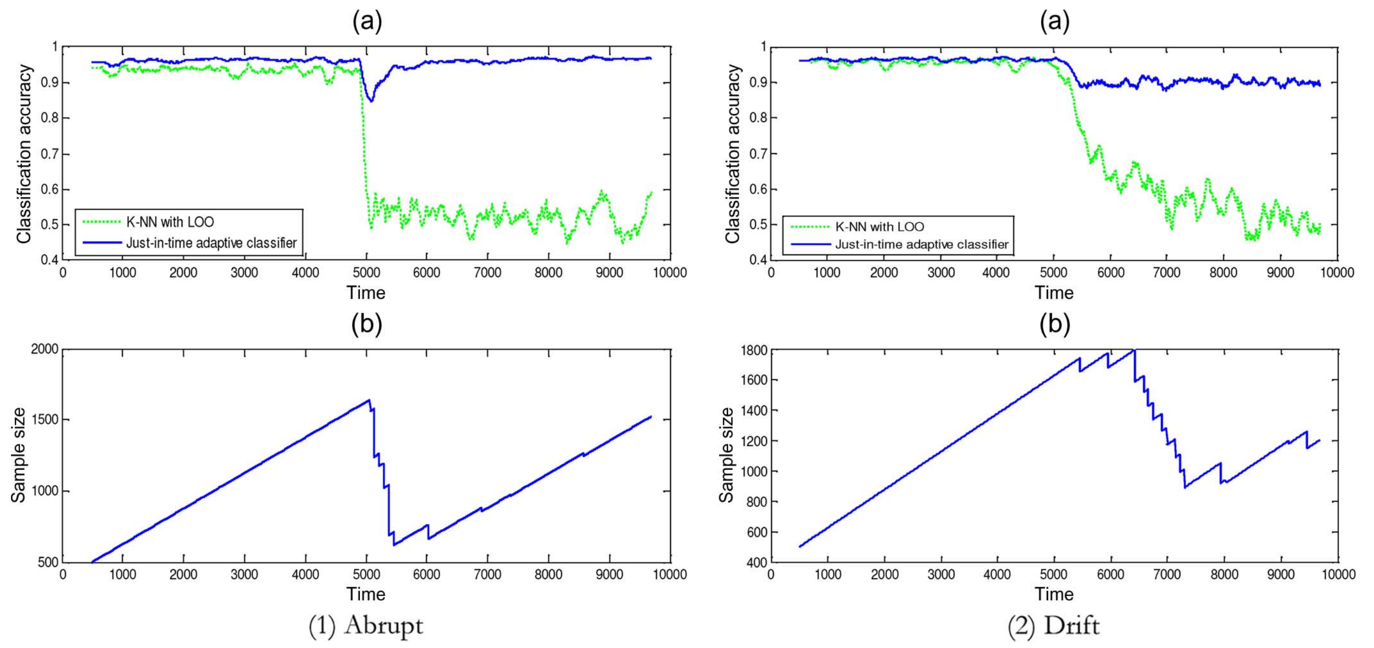

Fig. 7. Application D3. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN). (b) Number of samples in the KB of the JIT classifier during the experiment.

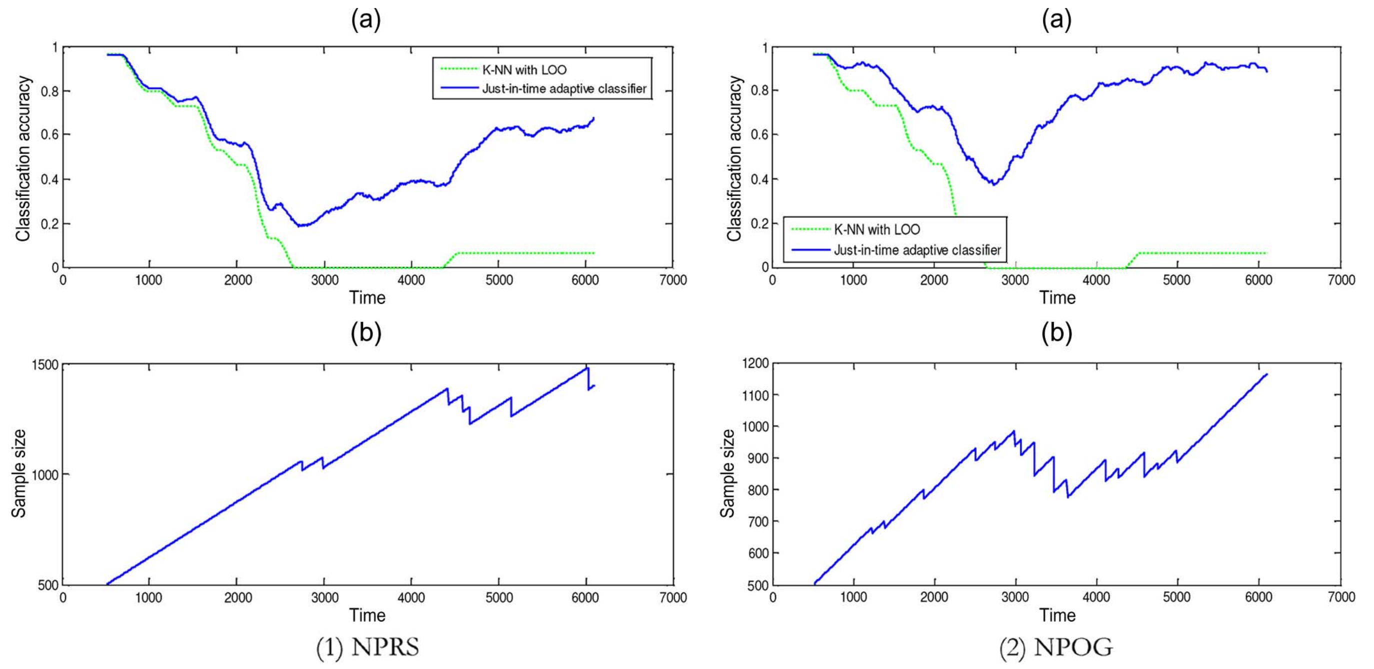

Fig. 8. Application D5. (a) Classification accuracy during the experiment (solid line: JIT classifier; dotted line: reference k-NN). (b) Number of samples in the KB of the JIT adaptive classifier during the experiment.

during the experiment. This implies a nonstationary behavior of the considered data set which, by inspecting the behavior of n and comparing it with previous applications, should be associated with a drift type of change. In this nonstationary case, the advantage of the use of the proposed JIT adaptive classifier is remarkable both for the respiratory signal (NRPS) and the electrooculography signal (NPOG).

Quantitative results of the JIT adaptive classifier and the reference k-NN with LOO in case of nonstationary conditions are summarized in Table III for the above presented applications. Results show the effectiveness of the proposed approach: JIT classifiers are always able to react to changes in the environment and provide a significant improvement in accuracy with respect to nonadaptive classifiers.

V. Conclusion

This paper presents a novel approach to design JIT adaptive classifiers working both in stationary and nonstationary conditions. Differently from the literature, the proposed JIT adaptive classifier updates, in a JIT fashion (i.e., only when it is really needed), its KB by exploiting information coming from the operational field after a change detection has been identified.

TABLE III
SIMULATION RESULTS OF THE JIT CLASSIFIER AND THE REFERENCE k-NN. LEGEND: $\mathrm{ACC}_{\mathrm{INIT}}$ AVERAGE ACCURACY (IN PERCENT) COMPUTED ON THE FIRST 1000 SAMPLES OF THE EXPERIMENT; $\mathrm{ACC}_{\mathrm{CHANGE}}$ AVERAGE ACCURACY (IN PERCENT) COMPUTED ON THE 1000 SAMPLES SUBSEQUENT TO THE CHANGE IN THE PROCESS; $\mathrm{ACC}_{\mathrm{END}}$ AVERAGE ACCURACY (IN PERCENT) COMPUTED ON THE LAST 1000 SAMPLES OF THE EXPERIMENT

		k-NN with LOO			**JIT Adaptive Classifier**		
		$\mathrm{Acc}_{\mathrm{init}}$ (%)	$\mathrm{Acc}_{\mathrm{change}}$ (%)	$\mathrm{Acc}_{\mathrm{end}}$ (%)	$\mathrm{Acc}_{\mathrm{init}}$ (%)	$\mathrm{Acc}_{\mathrm{change}}$ (%)	$\mathrm{Acc}_{\mathrm{end}}$ (%)
D1	Abrupt	73.5	61.4	60.8	73.9	61.3	72.8
	Drift	73.5	70.9	57.6	74.0	71.6	68.2
	Slow	73.5	62.3	61.7	73.6	63.5	73.3
	Oscil.	73.5	50.7	49.9	73.4	56.1	61.7
D2		58.7			65.8		
D3	Abrupt	93.2	72.2	52.1	96.3	92.6	96.2
	Drift	95.6	91.7	57.3	96.5	95.1	90.0
D5	NPOG	24.6			77.4		
	NPRS	24.2			52.6		

When no changes are detected, the adaptive classifier integrates fresh information into its KB so as to improve the classification accuracy. We demonstrate that the JIT classifier asymptotically converges to the Bayes classifier whenever a new stationary state can be granted. Moreover, the process behaves well when the environment evolves not too fast and allows the KB for containing representative training samples. Future research will address adaptive mechanisms able to provide acceptable performance even when the change dynamic of the process under investigation is faster than the tracking ability of the JIT adaptive classifier.

APPENDIX I

In stationary conditions, the classification error $\varepsilon(n)$ for a classifier trained with a training set Z_n of cardinality n can be expressed as [23]

$$\varepsilon(n) = \varepsilon_B + \varepsilon_a + \varepsilon_e \tag{4}$$

where ε_B, ε_a, and ε_e are the *Bayes* error, the *approximation* error (which depends on how the hypothesis space is close to the process generating the data), and the *estimation* error (which depends on the ability of the learning process to achieve the best hypothesis provided by the classifier), respectively. Consistency is granted provided that, as n increases, $\varepsilon(n) \to \varepsilon_B$. In the specific case of k-NN classifiers in stationarity conditions, we have consistency when

$$n \to \infty, k \to \infty \quad \text{and} \quad k/n \to 0 \tag{5}$$

hold.

A. No Transition: Stationary Case

Consistency for the JIT is granted whenever the process keeps operating in a stationary condition, since the classifier activates a simple k-NN classifier.

B. Transition: Moving From a Stationary State to a Different Stationary One

As discussed, in real applications, the case where the process, starting from a stationary situation, ends in a new stationary one is common (e.g., following a drift, an abrupt change or a combination of the two). In this case, the KB of the classifier Z_{new} is composed of two distinct contributions: n samples associated with the current stationary state (training set Z_n) and n_o samples associated with the previous history of the process (training set Z_{old}). Z_{old} contains information related to the past stationary state and that associated with the nonstationarity transition. Independently from the static classifier nature, the classification error can be expressed as

$$\varepsilon(n.n_o) = \varepsilon_B(Z_n) + \varepsilon_a(Z_n) + \varepsilon_e(Z_n) + \varepsilon^*(Z_{\mathrm{old}}) \tag{6}$$

where $\varepsilon^*(Z_{\mathrm{old}})$ represents a bias containing all effects induced by Z_{old}. It should be noted that such error is constant and, even when $n \to \infty$, it does not tend to 0. The consequence is that removal of Z_{old} from Z_{new} cannot be generally obtained and a bias error remains (the unique possibility for fully removing Z_{old} is associated with a strong abrupt change making it possible to separate old samples from new ones). Different heuristics for removing past data reduce $\varepsilon^*(Z_{\mathrm{old}})$ but general consistency cannot be granted, yet, the suggested JIT grants consistency with probability one, in the sense that $\varepsilon^*(Z_{\mathrm{old}})$ tends to 0 with probability 1, when $n \to \infty$. The proof of the statement relies on [44, Lemma 5.1], whose salient derivations are here presented to ease readability. Intuitively, the proof aims at identifying a neighbor of the sample to be classified containing only samples present in Z_n hence neglecting the presence of samples Z_{old}.

More formally, let $\overline{x} \in R^d$ be an input sample to be classified extracted from a continuous pdf for which the probability of extracting a specific sample is zero, let x_i be a generic training sample in Z_{new}, and let $\|x_i - \overline{x}\|$ be the Euclidian norm. Order sequence $\|x_i - \overline{x}\|$ according to increasing values and define $x^k(Z_n)$ to be the k-NN of $\overline{x}$ in Z_n and $x^k(Z_{\text{old}})$ the nearest neighbor in Z_{old}, respectively. Under hypotheses (5), the Lemma shows that $\|x^k(Z_n) - \overline{x}\|$ is monotone decreasing and tends to 0 with probability 1 when $n \to \infty$ (in other terms, the hypersphere centered at $\overline{x}$ of radius $\|x^k(Z_n) - \overline{x}\|$ shrinks monotonically to zero). Since we have a probability that $x^k(Z_{\text{old}}) \neq \overline{x}$, there exists a $d = \|x^k(Z_n) - \overline{x}\| > 0$, which does not depend on n or k. As such, when $n \to \infty$, we can identify a finite n^* and k^* for which $\|x^{k^*}(Z_{n^*}) - \overline{x}\| < d$ for all $n \geq n^*$ and the induced $k \geq k^*$. For all $n \geq n^*$, the JIT classifier will only operate with samples in Z_n and none from Z_{old}: consistency in probability is hence granted.

C. Transition: Nonstationarity

Finally, in situations where the process operates in nonstationary conditions, the classification problem is more complex. In particular, in drifting situations, the data generating process changes over time and consistency cannot be granted in general: the best possible action is to provide a classifier with a mechanism aiming at tracking the evolution change. As such, heuristics, such as the ones suggested in JIT classifiers, must be considered with performance and limits associated with the particular choice of the tracking mechanism. Appendix II shows how an adaptive selection of parameter T allows the JIT classifier for addressing the accuracy issue during the tracking phase.

Appendix II

The choice of parameter T is somehow critical since it defines the amount of samples to be considered representative of the current state of the process. A fixed value for it might be an inappropriate choice in many applications. In fact, if the dynamic of the change is rapid, we should consider a small T (i.e., only few data can be assumed as representative of the change) whereas a large T would be appropriate in correspondence of slow drifts. In other words, T needs to be adaptive.

At first, we should observe that the detection test has a memory and requires a window opened over past sample to detect a change. More specifically, if the change arises at time t_0, it will be detected only at time $\overline{t} > t_0$. This implies that, in case of abrupt variations, $\Delta n(t_0, \overline{t})$ samples between t_0 and $\overline{t}$ are representative of the change (e.g., constitute Z_n) and should not be discarded whereas samples acquired before t_0 belong to Z_{old} (Z_n and Z_{old} have been defined in Appendix I). Differently, in other scenarios (e.g., continuous drift), we cannot assure that all samples acquired between t_0 and $\overline{t}$ are representative of the current state of the process (since the process continuously changes); however, samples acquired before t_0 are obsolete.

A correct management of the KB has to then remove all obsolete (w.r.t. the current state of process) samples by eliminating, at least, all samples acquired before t_0 and keeping the $\Delta n(t_0, \overline{t})$ samples acquired between t_0 and $\overline{t}$. This allows the JIT classifier to minimize the accuracy loss introduced by change.

Change detection is here carried out by relying on the CI-CUSUM [22], which requires $T_{\min}$ (e.g., 500) samples to operate. Change is detected at time $\overline{t}$ when the difference $g(\overline{t}) = R(t) - m(t)|t = \overline{t}$ overcomes threshold h (automatically estimated by the test) and

$$R(t) = \sum_{\tau=1}^{t} \ln \frac{p_{\Theta^1}(x(\tau))}{p_{\Theta^0}(x(\tau))}$$

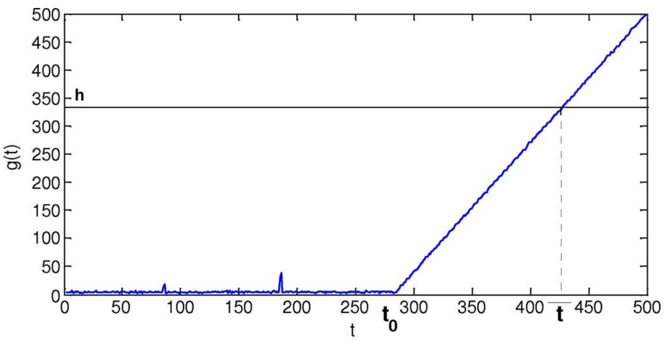

Fig. 9. Example of $g(t)$ over t (taken from application D5). The change starts at time t_0 and is detected at time $\overline{t}$.

is the cumulative sum of the log-likelihood ratio between the probability that a sample $x(\tau)$ belongs to a nonstationary process (hypothesis Θ^1) or to a stationary one (hypothesis Θ^0) and $m(t) = \min_{1 \leq \tau \leq t}(R(\tau))$. As a consequence, from the CI-CUSUM test, t_0 and $\overline{t}$ are $\overline{t} = \min\{t|g(t) > h\}$; $t_0 = \max\{t|g(t) = 0\}$.

The variable number of samples $T(\overline{t})$ to be considered in the KB is hence

$$T(\overline{t}) = \max\{T_{\min}, \Delta n(t_0, \overline{t})\}. \tag{B1}$$

It is interesting to see in Fig. 9 an example of evolution of $g(t)$ over t with respect to application D5. Before t_0, $g(t)$ (which assumes a zero value in the case of a stationary hypothesis) is only subject to statistical fluctuations; no changes are detected since $g(t)$ is below threshold h (the solid horizontal line). Then, the process starts drifting (time t_0) and only at time $\overline{t}$ it is detected, i.e., $g(t)$ overcomes h.

In an operational framework, one has to then start from t_0 and go back in time to identify $\overline{t}$ by inspecting those values of $g(t)$ above the "background noise" associated with a stationary condition.

References

[1] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 238–250, Aug. 1996.
[2] L. E. Doherty, "A computer based adaptive learning process," in *Proc. Int. Conf. Knowl.-Based Intell. Electron. Syst.*, Zagreb, Croatia, 1998, vol. 3, pp. 303–306.
[3] Q. Jackson and D. A. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 12, pp. 2664–2679, Dec. 2001.
[4] M. M. Rizki, M. A. Zmuda, and L. A. Tamburino, "Evolving pattern recognition systems," *IEEE Trans. Evolut. Comput.*, vol. 6, no. 6, pp. 594–609, Dec. 2002.
[5] N. Iwayama, K. Akiyama, and K. Ishigaki, "Hybrid adaptation: Integration of adaptive classification with adaptive context processing," in *Proc. Int. Workshop Frontiers Handwriting Recognit.*, Aug. 2002, pp. 169–174.
[6] N. Kasabov, "Evolving connectionist systems for adaptive learning and knowledge discovery: Methods, tools, applications," in *Proc. Int. Conf. Neural Inf. Process.*, Singapore, Nov. 2002, vol. 2, pp. 590–595.
[7] M. R. Azimi-Sadjadi, D. Yao, A. A. Jamshidi, and G. J. Dobeck, "Underwater target classification in changing environments using an adaptive feature mapping," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1099–1111, Sep. 2002.
[8] J. Wang, M. R. Azimi-Sadjadi, and D. Reinke, "A temporally adaptive classifier for multispectral imagery," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 159–165, Jan. 2004.

[9] F. Virili and B. Freisleben, "Nonstationarity and data preprocessing for neural network predictions of an economic time series," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Como, Italy, Jul. 2000, vol. 5, pp. 129–134.

[10] P. Li, W. Melvin, M. Wicks, P. Antonik, and H. Schuman, "Knowledge-based space-time adaptive processing," in *Proc. IEEE Nat. Radar Conf.*, Syracuse, NY, 1997, pp. 372–377.

[11] H. Shah-Hosseini and R. Safabakhsh, "Pattern classification by the time adaptive self-organizing map," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, Jounieh, Lebanon, Dec. 2000, vol. 1, pp. 495–498.

[12] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "A fuzzy artmap nonparametric probability estimator for nonstationary pattern recognition problems," *IEEE Trans. Neural Netw.*, vol. 6, no. 6, pp. 1330–1336, Nov. 1995.

[13] L. Rutkowski, "Adaptive probabilistic neural networks for pattern classification in time-varying environment," *IEEE Trans. Neural Netw.*, vol. 15, no. 4, pp. 811–827, Jul. 2004.

[14] A. Kuh, "Performance of analog neural networks subject to drifting targets and noise," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Washington, DC, Jul. 1999, vol. 4, pp. 2406–2409.

[15] A. Kuh, "Comparison of tracking algorithms for single layer threshold networks in the presence of random drift," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 640–649, Mar. 1997.

[16] M. A. Elbestawi, N. K. Sinha, R. Lingarkar, and L. Liu, "Knowledge-based adaptive computer control in manufacturing systems: A case study," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 3, pp. 606–618, May/Jun. 1990.

[17] W. H. Land, L. Albertelli, Y. Titkov, P. Kaltsatis, and G. Seburyano, "Evolution of neural networks for the detection of breast cancer," in *Proc. IEEE Int. Joint Symp. Intell. Syst.*, Rockville, MD, 1998, pp. 34–40.

[18] P. Robertson and J. M. Brady, "Adaptive image analysis for aerial surveillance," *IEEE Intell. Syst. Their Appl.*, vol. 14, no. 3, pp. 30–36, May–Jun. 1999.

[19] S. C. Tan and C. P. Lim, "Condition monitoring and fault prediction via an adaptive neural network," in *Proc. TENCON*, Kuala Lumpur, Malaysia, Sep. 2000, vol. 1, pp. 13–17.

[20] T. E. Boult, R. J. Micheals, X. Gao, and M. Eckmann, "Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings," *Proc. IEEE*, vol. 89, no. 10, pp. 1382–1402, Oct. 2001.

[21] A. Siddique, G. S. Yadava, and B. Singh, "Applications of artifcial intelligence techniques for induction machine stator fault diagnostics: Review," in *Proc. IEEE Int. Symp. Diag. Electr. Mach. Power Electron. Drives*, Stone Mountain, GA, Aug. 2003, pp. 29–34.

[22] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers—Part I: Detecting nonstationarity changes," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1145–1143, Jul. 2008.

[23] C. Alippi and P. Braione, "Classification methods and inductive learning rules: What we may learn from theory," *IEEE Trans. Syst. Man Cybern.*, vol. 36, no. 5, pp. 649–655, Sep. 2006.

[24] H. Chernoff, *Sequential Analysis and Optimal Design*. Philadelphia, PA: SIAM, 1972.

[25] T. M. Cover and R. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[26] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.

[27] C. Stone, "Consistent nonparametric regression," *Ann. Statist.*, vol. 8, pp. 1348–1360, 1977.

[28] K. Fukunaga and L. Hostetler, "Optimization of k nearest neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. 19, no. 3, pp. 320–326, May 1973.

[29] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[30] P. A. Lachenbruch and M. R. Mickey, "Estimation of error rates in discriminant analysis," *Technometrics*, vol. 10, pp. 1–10, 1968.

[31] M. Keans and D. Ron, "Algorithmic stability and sanity check bounds for leave-one-out cross validation bounds," *Neural Comput.*, vol. 11, no. 6, pp. 1427–1453, 1999.

[32] L. P. Devroye and T. J. Wagner, "Distribution-free inequalities for the deleted and holdout error estimates," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, pp. 601–604, Sep. 1979.

[33] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 515–516, May 1968.

[34] G. W. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 431–433, May 1972.

[35] D. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst. Man Cybern.*, vol. 2, no. 3, pp. 408–421, Jul. 1972.

[36] R. S. Poulsen, B. K. Bhattacharya, and G. T. Toussaint, "Application of proximity graphs to editing nearest neighbor decision rule," in *Proc. Int. Symp. Inf. Theory*, Santa Monica, CA, 1981, pp. 1–25.

[37] C. Zonghai, Z. Haitao, L. Ming, and X. Wei, "Adaptive control method for nonlinear time-delay processes," *J. Syst. Eng. Electron.*, vol. 18, pp. 566–576, 2007.

[38] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," in *Proc. 17th Conf. Pattern Recognit.*, Cambridge, U.K., Aug. 2004, vol. 3, pp. 430–433.

[39] E. L. Russel, L. H. Chiang, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. New York: Springer-Verlag, 2001.

[40] I. V. Nikiforov, "A generalized change detection problem," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 171–187, Jan. 1995.

[41] C. Blake and C. Merz, UCI Machine Learning Databases Repository, Dept. Inf. Comput. Sci., Univ. California-Irvine [Online]. Available: http://www.ics.uci.edu/~mlearn/databases/statlog/, Last accessed on: May 2008

[42] M. Bollano, R. Piagge, A. Charai, and D. Narducci, "Experimental evidence and computational analysis of the electronic density modulation induced by gaseous molecules at Si(001) surfaces upon self-assembling organism monolayer," *Appl. Surface Sci.*, vol. 175–176, pp. 379–385, 2001.

[43] D. Narducci, P. Bernardiello, and M. Oldani, "Investigation of gas-surface interactions at self-assembled silicon surfaces acting as gas sensor," *Appl. Surface Sci.*, vol. 212–213, pp. 491–496, 2003.

[44] J. Kohlmorgen, K.-R. Müller, J. Rittweger, and K. Pawelzik, "Identification of nonstationary dynamics in physiological recordings," *Biol. Cybern.*, vol. 83, no. 1, pp. 73–84, 2000.

Cesare Alippi (SM'94–F'06) received the Dr.Ing. degree in electronic engineering *(summa cum laude)* and the Ph.D. degree in computer engineering, both from Politecnico di Milano, Milano, Italy, in 1990 and 1995, respectively.

He has completed research work in computer sciences at the University College London, London, U.K., and the Massachusetts Institute of Technology, Cambridge. Currently, he is a Full Professor of Information Processing Systems at the Politecnico di Milano. His research interests include application-level analysis and synthesis methodologies for embedded systems, computational intelligence, and wireless sensor networks. His research results have been published in more that 120 technical papers in international journals and conference proceedings.

Dr. Alippi is the Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Manuel Roveri received the Dr.Eng. degree in computer science engineering from the Politecnico di Milano, Milano, Italy, in June 2003, the M.S. degree in computer science from the University of Illinois at Chicago, Chicago, in December 2003, and the Ph.D. degree in computer engineering from Politecnico di Milano, Milano, Italy, in May 2007.

His research interests include computational intelligence, adaptive algorithms, and wireless sensor networks.