Original Article

Institution of MECHANICAL ENGINEERS

Journal of RISK AND RELIABILITY

Proc IMechE Part O: J Risk and Reliability
2021, Vol. 235(1) 33–49

DOI: 10.1177/1748006X20946529
journals.sagepub.com/home/pio
SAGE

A method for fault diagnosis in evolving environment using unlabeled data

Yang Hu[1], Piero Baraldi[2], Francesco Di Maio[2], Jie Liu[3] and Enrico Zio[2,4,5]

Abstract

Industrial components and systems typically operate in an evolving environment characterized by modifications of the working conditions. Methods for diagnosing faults in components and systems must, therefore, be capable of adapting to the changings in the environment of operation. In this work, we propose a novel fault diagnostic method based on the compacted object sample extraction algorithm for fault diagnostics in an evolving environment from where unlabeled data are collected. The developed diagnostic method is shown able to correctly classify data taken from synthetic and real-world case studies.

Keywords

Concept drift, drift detection, fault diagnostics, α shape reconstruct, evolving environment, bearing

Date received: 5 October 2019; accepted: 8 July 2020

Introduction

Prognostics and health management (PHM) is an interdisciplinary field of research and application, whose aim is to use the available data, information and knowledge about the system environmental, operational and usage conditions to detect its degradation, diagnose its faults, predict the future evolution of its health state and remaining useful life and proactively manage its failures.[1–5] In the overall PHM process, fault diagnostics is an essential step for identifying the source of system failures and, therefore, providing the information needed for an effective condition-based maintenance. The identification of the cause of the fault allows reducing the mean time to repair (MTTR) and increase system availability.[6]

In the last decades, a number of machine learning methods have been developed for the classification of the type of fault causing the malfunctioning of components and systems.[7] The proposed methods have been reported to achieve satisfactory performance in several works. However, the industrial applicability of these methods is limited by the fact that the training data available to build the empirical model for the diagnosis typically do not include all the working conditions that the components and systems may experience during life. As a result, if the diagnostic model is used in working conditions that differ from those considered to train the model, its performance maybe unsatisfactory. Such problem occurs for components such as bearings, gears, alternators, shafts, pumps, and so on, operating in an evolving environment characterized by continuous or periodic modifications of the working conditions, for example, applied loads of the operational environmental conditions.[8,9]

A possible solution to this problem is to periodically update the diagnostic model by adaptively training it on data collected in the newly experienced working conditions. For this, the idea of incrementally learning the information that progressively becomes available has been proposed.[10–13] For example, in Razavi-Far

[1]Science and Technology on Complex Aviation System Simulation Laboratory, Beijing, China
[2]Department of Energy, Politecnico di Milano, Milano, Italy
[3]School of Reliability and System Engineering, Beihang University, Beijing, China
[4]Aramis Srl, Milano, Italy
[5]MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France

Corresponding author:
Yang Hu, Science and Technology on Complex Aviation System Simulation Laboratory, Beijing 100076, China.
Email: yang.hu@polimi.it

et al.,[10] the authors proposed an ensemble approach where a new model is developed and added to the ensemble each time a new set of data becomes available. In Ben-David and Blitzer[14] and Daume and Marcu,[15] the problem had been addressed in the framework of "domain adaptation," assuming that labeled patterns become available during operation: this requirement can significantly limit the possibility of using the approach in practical fault diagnostic applications where the identification of the class of the fault causing the malfunctioning is often not feasible or very expensive and time consuming.

Another possible solution is to online update the parameters of the diagnostic model. This approach was applied in Dong and He[16] and Li et al.[17] to hidden Markov models (HMMs) and in Li and Xu[18] and Karami and Wang[19] to Gaussian mixture models (GMMs). In Dong and He,[16] the parameters of the HMM model were online updated using a modified forward–backward training algorithm, which allows predicting the system's useful remaining life considering the novel acquired data. In Li et al.,[17] the authors proposed a method for updating the matrix of the state transition probabilities of the HMM model using the online collected degradation data. In Li and Xu[18] and Karami and Wang,[19] GMM was used for modeling the component degradation model and Bayesian filtering methods were applied for online updating the model parameters. Notice that these works propose approaches that are designed to manage gradual modification of the environment and are not able to handle the situation in which the working conditions suddenly and abruptly change.

In this work, we present the development of a systematic fault diagnostics approach with continuous and automatic updating of the diagnostics model, without the requirement of new labeled data. The proposed approach is based on the modification of a "compacted object sample extraction (COMPOSE)" algorithm for incremental learning developed in the domain of streaming data learning.[20,21] COMPOSE allows obtaining satisfactory classification performances in presence of concept drifts, that is, situations in which the statistical properties of the class of the patterns change over time in an unforeseen way.[22–24] It has been applied with success to the classification of synthetic and real-world data sets.[20] The key idea of COMPOSE is to aggregate the labeled training data with the unlabeled new data and to perform a shrinkage of the obtained data set in order to identify a core region representing the trend of the concept drift. The main advantages of COMPOSE with respect to other incremental learning approaches are that (a) model updating does not require the knowledge of the class of the new available patterns and (b) it can be applied to various types of concept drifts.

In Hu et al.,[25] the COMPOSE method has been adapted to fault diagnostics. However, since COMPOSE is designed for stream data learning problems characterized by a large amount of patterns becoming available in a short time, it needs to be modified for applications in the domain of fault diagnostics where the number of patterns becoming available is typically very limited. Indeed, in this latter situation, the application of COMPOSE each time a new batch of data becomes available causes an over-shrinkage of the training set, and, thus, a fast reduction of the diagnostic model classification performance.

To overcome this limitation of COMPOSE, the main novelty of this article is the development of a diagnostic approach which includes a new concept drift detection module, based on the construction of α shape surfaces.[26,27] The objective of the module is to identify when it is necessary to update the classification model, in order to reduce the training set over-shrinkage caused by the COMPOSE algorithm and the computation burden of updating the diagnostics model. A second novelty of the work is an automatic procedure for setting the internal parameters of COMPOSE, in such a way that the size of the training set remains stable.

The proposed concept drift module and the overall fault diagnostic method are verified with respect to three data sets containing concept drifts: (1) two synthetic data sets made by artificial unimodal Gaussian and irregular distributions and (2) a real-world bearing vibration data set taken from Case Western Reserve University.[28] The concept drift module performances are compared to those of a semi-parametric log-likelihood (SPLL) method. In all the case studies, different types of concept drift and data distributions are considered.

The remainder of this article is organized as follows: Sections "Problem statement" and "The proposed method for fault diagnostics in an evolving environment" illustrate the problem of fault diagnostics in an evolving environment; section "Concept drift detection" discusses the method for concept drift detection; the proposed method for identifying the new training set to be used for model updating is introduced in section "Construction of the new training set"; sections "Synthetic case studies" and "Real-world case study" show the performance of the proposed approach on the three case studies; and section "Conclusion" summarizes the whole article and puts forward some future work for improvement of the proposed method.

Problem statement

Fault diagnostics is typically considered a supervised classification problem.[29,30] Signal measurements are given as input to an empirical classifier, which provides output of the class of the fault. The classifier is developed by considering a training set of n_T patterns, $T = \{\boldsymbol{x}_i^T, l_i^T\}, i = 1, 2, \ldots, n_T$, formed by the vector of the signal measurements, $\boldsymbol{x}_i^T$, and corresponding class label of the fault, l_i^T. Once the empirical classifier has been trained, it can be used to classify a test set of n_C unlabeled patterns, $C = \{\boldsymbol{x}_i^C\}, i = 1, 2, \ldots, n_C$, formed by the vectors of the signal measurements, $\boldsymbol{x}_i^C$, whose true class labels, l_i^C, are unknown.

Pattern classification is tacitly based on the hypothesis that the training set T and the test set C are similar, that is, their patterns are originated from similar probability distributions. However, this hypothesis does not hold when fault diagnostics is performed in an evolving environment, since modifications of the working conditions cause modifications of the signal measurements, so that patterns of the test set are originated by probability distributions different from those of the training set. Therefore, the performance of the diagnostic model trained using $\boldsymbol{x}_i^T$ can deteriorate as the component or system operational conditions change in time.

To overcome this problem, the main objectives of this work are

1. To develop a method for detecting whether there has been a modification of the working conditions causing a deterioration of the diagnostic model performance. According to Kuncheva and Faithfull,[31] this problem will be referred to as the problem of concept drift detection, since the modification of the working conditions causes a drift in the distributions of the test patterns, $\boldsymbol{x}_i^C$, with respect to those of the training patterns, $\boldsymbol{x}_i^T$.
2. To develop a method for updating the classification model once a drift has been detected.

Since the labeling of the test patterns is often not feasible or very expensive and time consuming in practical fault diagnostic applications, the two problems will be addressed assuming that the true class labels, l_C^i, of the patterns, $\boldsymbol{x}_C^i$, of the test set, C, are not known.

The proposed method for fault diagnostics in an evolving environment

The overall method for fault diagnostics in an evolving environment is based on the following three modules (Figure 1):

1. A fault classifier (FC).
2. A concept drift detector (CDD).
3. A module for the identification of the patterns to be used for retraining the FC once a concept drift has been detected.

Let us assume that an FC has been developed using the labeled training patterns, T. Once a new batch of unlabeled patterns, $C = \{\boldsymbol{x}_i^C\}, i = 1, 2, \ldots, n_C$, is collected, the patterns are pre-classified by the FC, that is, a tentative class label $\tilde{l}_i^C$ is assigned to each pattern. Then, C is sent to the CDD: in case of no concept drift detection, the classification labels are confirmed, whereas, in case of concept drift detection, the FC is updated. This requires the construction of a new training set, which will be used for retraining the FC. Finally, the updated classifier can be used for the classification of the patterns in C.

In this work, we focus our attention on the CDD (section "Concept drift detection") and on the method for the construction of the training set which is used for retraining the FC (section "Construction of the new training set"), whereas the discussion of classification algorithms for fault diagnostics is outside the scope of this work (the interested reader can refer to the literature[32–35] for examples of application of empirical classification algorithms to fault diagnostic problems).

Concept drift detection

The problem of concept drift detection has been extensively studied for its potential application in very different areas such as quality control,[36] medical condition monitoring[37] and market analysis.[38,39] In the framework of fault diagnostics, the objective of concept drift detection is to verify whether the statistical properties of the test patterns are significantly different from those of the training patterns used to develop the diagnostic model.

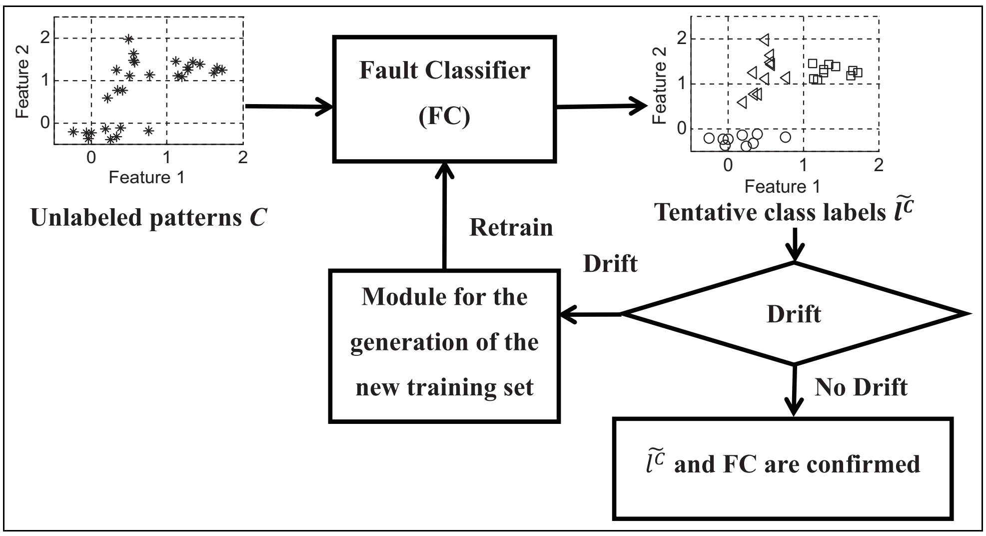

Figure 1. Sketch of overall fault diagnostic method.

Time windows–based concept drift detection methods

Common approaches to concept drift detection are based on the analysis of the statistical distributions of the data in consecutive time windows. Methods based on the use of the Kullback–Leibler[31,40] divergence and the Hotelling's T-squared distribution[41] are typically employed for this aim. In Kuncheva and Faithfull,[31] the authors have shown the effectiveness of an approach based on the use of the SPLL value, which provides more satisfactory performances than those employing the Kullback–Leibler divergence and the Hotelling's T-squared distribution. The SPLL value is defined by

$$SPLL(W_1, W_2) = \frac{1}{M_2} \sum_{x \in W_2} \max_{i=1}^{N} (P(\boldsymbol{x}|W_1)) \quad (1)$$

where W_1 and W_2 are the two consecutive time windows of data, M_2 is the number of patterns in the time window W_2, $\boldsymbol{x}$ is the vector of the signal measurements in the W_2 time window and $P(\boldsymbol{x}|W_1)$ is the likelihood of the vector $\boldsymbol{x}$ given the data in the time window W_1. SPLL represents the upper bound of the likelihood of the data in the time window W_2 given the data in the time window W_1. Large SPLL values indicate that the statistical distributions of the data in W_1 and W_2 are similar, whereas small SPLL values indicate the occurrence of a concept drift. The threshold to be used for the concept drift detection is decided by observing the receiver operating characteristic (ROC) curve[42] and choosing the threshold value which allows achieving the best tradeoff between missed and false detections.[31]

The SPLL method tends to fail when it is applied to data characterized by very irregular probability distributions, such as shown in Figure 2, for which the SPLL value is very similar before (left) and after (right) the occurrence of the concept drift. Since these situations are expected to be common in fault diagnostics, a novel method for concept drift detection is proposed in following section.

The proposed concept drift detection method

The proposed method for concept drift detection is based on the use of the α shape reconstruction technique,[26] which produces a surface enveloping a dense unorganized set of multidimensional patterns. A detailed description of the method can be found in the literature.[26,43,44] Here, we focus on the choice of the parameter α, which controls the level of details and roughness of the enveloping surface. According to Edelsbrunner and Mucke,[43] α can be seen as the radius of a hyper-spherical spoon, which is used to "carve out" around the data set.

Figure 3 shows the enveloping surfaces obtained by applying the α shape reconstruction technique to bi-dimensional patterns randomly sampled from the character α and considering different values of the parameter α. When too low values of parameter α are used, such as in case 2, the obtained surface is characterized by a very irregular shape with voids between the patterns, whereas, when too large values are used, such as in cases 4 and 5, the surface becomes very rough and it does not provide enough detailed information on the data set.[1]

According to the analysis in Guo et al.[26] and Edelsbrunner and Mucke,[43] the value of the parameter α should be set considering the density of the data

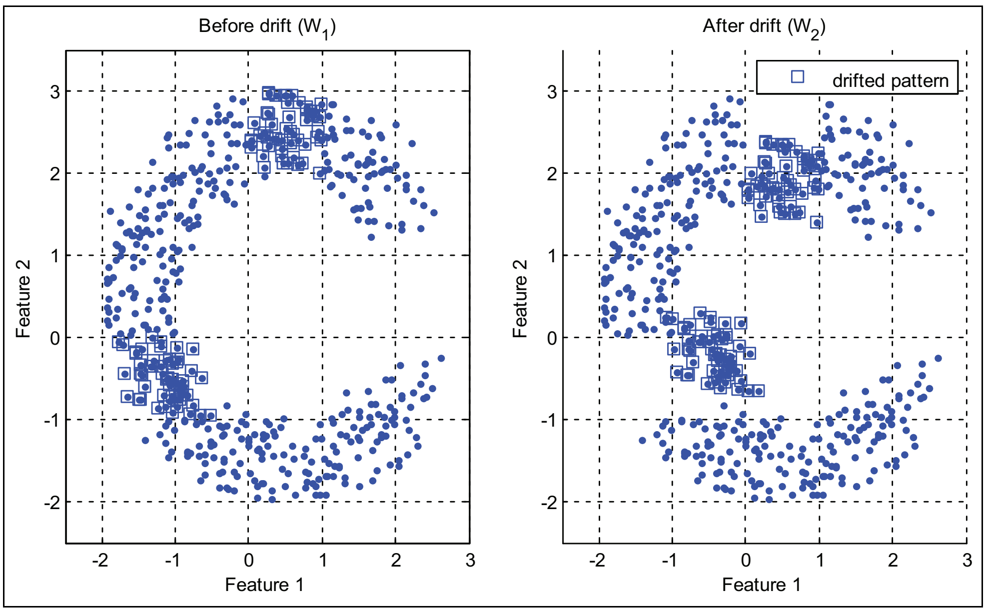

Figure 2. Data distributions before (left) and after (right) the concept drift occurrence.

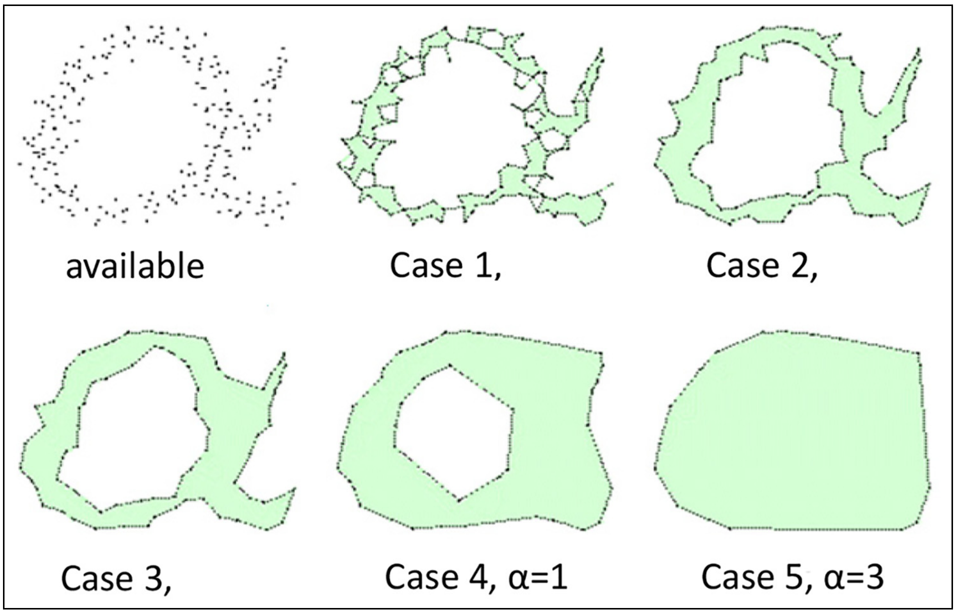

Figure 3. Top right: available patterns, cases 1–5: α shape reconstruction surfaces obtained for different values of the parameter α (from Edelsbrunner and Mucke[43]).

patterns: larger the density, smaller the α value. In this work, α is set considering the average minimum distance among the patterns

$$\alpha = \frac{1}{n}\sum_{i}^{n}\left(\min_{j=1, j\neq i}^{n}(d_{ij})\right) \tag{2}$$

where d_{ij} is the Euclidean distance between patterns x_i and x_j and n is the number of patterns.

The basic idea behind using the α shape reconstruction technique for concept drift detection is that the drifted patterns are expected to be outside the α shape surface enveloping the training set. Thus, in case of drift, the volume of the α shape surface enveloping the union of the training patterns and the new batch of test patterns is expected to be larger than that enveloping only the training patterns.

Given the training set $T = \{\boldsymbol{x}_i^T\}, i = 1, 2, \ldots, n_T$, the detection of a concept drift in a new batch of test patterns, $C = \{\boldsymbol{x}_j^C\}, j = 1, 2, \ldots, n_C$, is based on the following procedures:

1. Set the detection threshold $f > 1$.
2. Initialize the subset of the drifted patterns, $D = \emptyset$, and that of the non-drifted patterns, $ND = \emptyset$.
3. Set the α value by applying equation (2) to the patterns of training set, T.
4. Build the α shape surface S_T of the training set and compute volume VT.
5. Set the indicator of the pattern in the test batch, j, equal to 1.
6. Build a new data set, A_j, as the union of the training set T and the jth pattern of the test set batch, C, that is, $A_j = \{T \cup \boldsymbol{x}_j^C\}$.
7. Build the α shape surface S_{A_j} and compute its volume V_{A_j}.
8. Find the ratio $R_j = V_{A_j}/V_T$; if $R_j > f$, put $\boldsymbol{x}_j^C$ into the subset of the drifted patterns, that is, $D = D \cup \boldsymbol{x}_j^C$, otherwise, put $\boldsymbol{x}_j^C$ into the subset of the non-drifted patterns, that is, $ND = ND \cup \boldsymbol{x}_j^C$.
9. Set $j = j + 1$ and go to step 6 until all the patterns in C are tested.

Once the procedure has been applied to all the patterns of the test batch, it provides the two subsets of the drifted, D, and non-drifted, ND, patterns.

Figure 4 shows an example of application of the α shape reconstruction technique for concept drift detection. The left figure shows the α shape surface of the training set, T; the figure in the middle shows a case in which the test pattern (square) is inside the α-shape surface of T and, thus, it leaves unmodified the area of the α-shape surface; the right figure shows a case in which $\boldsymbol{x}_j^C$ is outside the α-shape surface of the training pattern, and, thus, causes an increase in the area.

Construction of the new training set

According to the scheme of Figure 1, once a concept drift has been detected, the diagnostics model is updated. This requires the construction of a new training set and the development of a new classification model. In this work, the construction of the new training set is inspired by the COMPOSE algorithm. The basic assumption behind COMPOSE is that the core region of the training data overlaps, at least partially, with a part of the drifted data. According to Dyer et al.,[20] this assumption is met when the concept drift is

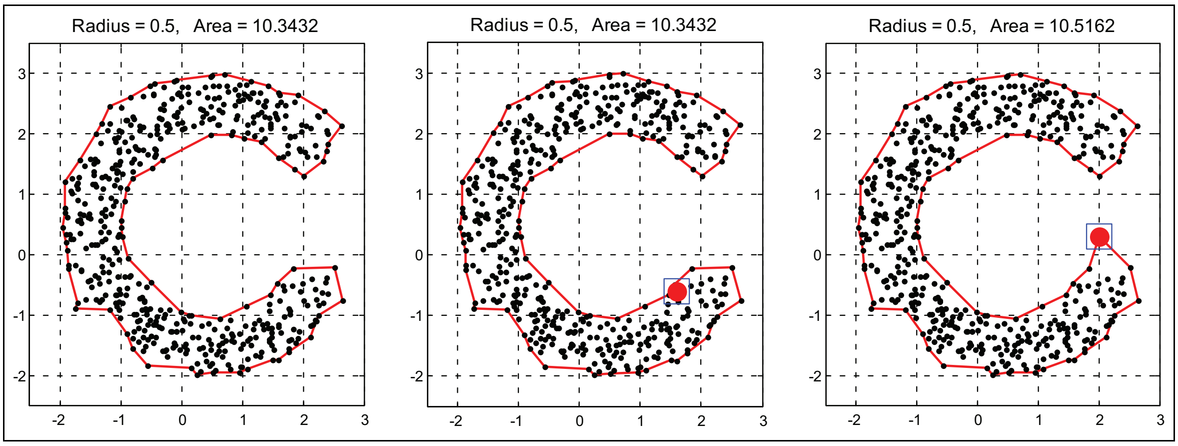

Figure 4. Sketch of α shape–based drift detection method.

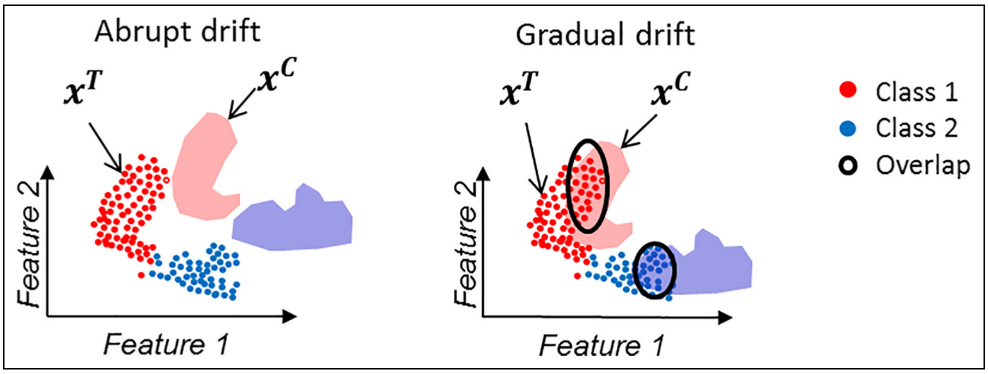

Figure 5. Examples of abrupt (left) and gradual (right) drifts.

gradual, that is, the training set T and the test set C have some overlapping area, as shown in Figure 5. Examples of gradual drifts in fault diagnostic applications are encountered in cases of variations of the working conditions due to slow degradation processes in the component of system or to seasonal variations in the external environment.

Assuming that an FC has been already trained using the labeled data in T and that a concept drift has been identified in the test set C, the construction of a new training set is based on the following steps:

1. An aggregate data set formed by the labeled patterns of the training set T and those of C, with corresponding labels assigned by the empirical FC, is built.
2. The α shape surface reconstruction is applied to find the surface boundary of each class in the aggregated data set; parameter α allows regulating the level of details and roughness of the surfaces.
3. The core regions of each class are identified by performing a shrinkage of the obtained class surface boundaries. Then, the patterns outside the surface of the α shape are removed.
4. The new training set is formed by the labeled patterns in the core regions identified in step 3.

Once the new training set has been obtained, it is used to train a new classifier which substitutes the old one. The procedure is entirely repeated each time a new concept drift is detected in a new batch of unlabeled patterns.

The novelties of the proposed procedure with respect to that in the COMPOSE algorithm are

1. It is applied only when a concept drift is detected in the test set C and not, as in the COMPOSE algorithm, each time a set of unlabeled patterns is collected.
2. The parameter α is set according to equation (2) in section "Time windows–based concept drift detection methods."
3. The parameter CP which regulates the shrinkage percentage is set to

$$CP = 1 - \frac{n_T}{n_A} \tag{3}$$

where n_T is the number of the patterns in the training data set and n_A is the number of the patterns in the aggregated data set. This allows maintaining the number of patterns in T unchanged after shrinkage, avoiding the phenomena of under and over-shrinkage.

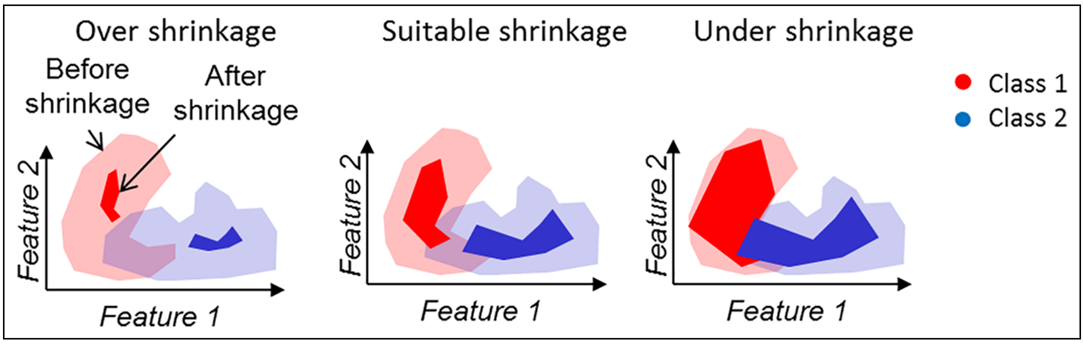

Figure 6. Examples of different degrees of shrinkage of a two-class data set.

Under-shrinkage is obtained when too small *CP* values are used and results in too large core regions with consequent overlapping of core regions of different classes (Figure 6). This can deteriorate the performance of the classifier, since it causes the inclusion of patterns with wrong label in the new training set. Over-shrinkage of the α-surfaces arises when too large *CP* values are used, with consequent elimination of useful patterns from the training set found in step 2, and, thus, inability of updating the algorithm in order to follow the concept drift. Notice that the proposed strategy, differently from that in Dyer et al.,[20] allows using different *CP* values for different classes of patterns.

The details of the overall algorithm and its flowchart are given in Appendix 1.

Synthetic case studies

In this section, we verify the proposed method considering two synthetic case studies. The first data set contains data sampled from bi-dimensional Gaussian distributions, whereas the second from non-Gaussian distributions.

The performance of the proposed diagnostic approach has been compared with those of the

1. Original COMPOSE algorithm.[20] Parameters α and *CP* are set to 0.6, and the classifier is updated each time a new batch of data becomes available.
2. Original COMPOSE algorithm combined with the SPLL algorithm for concept drift detection. The classifier is updated when SPLL detects the concept drift.

In both cases, the K-nearest neighbor (KNN) classifier[45] has been used as base classifier and the class of the patterns that becomes available during the experiments in the evolving environments is assumed to be unknown.

A fully supervised KNN classifier in which we unrealistically assume that labeled patterns become available each time step and are immediately used for retraining the classifier is also considered to provide an upper bound of the classification performance. This latter approach will be referred to as reference approach.

Synthetic case study I

Data are assumed to become available in batches of N_batch = 100 patterns collected every 0.01 arbitrary time units.[15] The patterns of a given class are sampled from bi-dimensional Gaussian distributions whose mean and standard deviation, μ and σ, are reported in Table 1. The presence of an evolving environment is reproduced by periodically changing the mean and standard deviation of the distributions. All the patterns provided to the diagnostic system are unlabeled except those of the initial batch collected at time $t = 0$ and of the batch collected at time $t = 0.6$, when patterns of a new class appear. The batches are formed by patterns of two classes until time $t = 0.6$ and of three classes after time $t = 0.6$. The simulation is based on five cycles. In each cycle, there are 20 consecutive batches in which the data distributions are slowly changing (due to the presence of a concept drift) followed by 10 consecutive batches in which the data are sampled from the same distributions (no concept drift). The purpose of this setting is to simulate the operation of a component or system characterized by phases with changing load followed by phases with constant load.

The experiment has been repeated 50 times, each time with a new simulation of the batches of data, using the distributions in Table 1.

Figure 7 compares the accuracy (fraction of patterns correctly classified in a batch) of all the approaches. Notice that

- The median values of the accuracy in the three approaches over the 50 repetitions of the experiment (dotted lines) do not show significant differences.
- The 90% confidence intervals of the proposed approach, which are computed by the 50 repetitions of experiments, are remarkably smaller. This indicates that the proposed approach is more adaptable and robust than the other two. In particular, the original COMPOSE algorithm is characterized

Table 1. Parameters of the Gaussian distributions used to sample the data in the different batches.

Class	Batches with concept drift				Batches without concept drift			
	Cycle 1 (batches 1–30)							
	$t \in [0, 0.2)$ (batches 1–20)				$t \in [0.2, 0.3)$ (batches 21–30)			
	μ_x	μ_y	σ_x	σ_y	μ_x	μ_y	σ_x	σ_y
C1	$2 - 5t$	5	1.5	$5 - 5t$	1	5	1.5	4
C2	$5 - 5t$	8	$5 - 15t$	$1.5 + 2.5t$	4	8	2	2
C3	NA	NA	NA	NA	NA	NA	NA	NA
	Cycle 2 (batches 31–60)							
	$t \in [0.3, 0.5)$ (batches 31–50)				$t \in [0.5, 0.6)$ (batches 51–60)			
C1	1	$5 - 10(t - 0.3)$	$1.5 + 7.5(t - 0.3)$	3	1	3	3	3
C2	$4 + 20(t - 0.3)$	8	2	2	8	8	2	2
C3	NA	NA	NA	NA	NA	NA	NA	NA
	Cycle 3 (batches 61–90)							
	$t \in [0.6, 0.8)$ (batches 61–80)				$t \in [0.8, 0.9)$ (batches 81–90)			
C1	1	$3 - 5(t - 0.6)$	$3 - 10(t - 0.6)$	$3 - 10(t - 0.6)$	1	2	1	1
C2	8	$8 - 20(t - 0.6)$	$2 - 5(t - 0.6)$	$2 + 10(t - 0.6)$	8	4	1	4
C3	5	$5 + 15(t - 0.6)$	$1 + 5(t - 0.6)$	$1 + 5(t - 0.6)$	5	8	2	2
	Cycle 4 (batches 91–120)							
	$t \in [0.9, 1.1)$ (batches 91–110)				$t \in [1.1, 1.2)$ (batches 111–120)			
C1	$1 - 5(t - 0.9)$	$2 + 15(t - 0.9)$	$1 + 15(t - 0.9)$	1	0	5	4	1
C2	8	$4 + 20(t - 0.9)$	1	$4 - 10(t - 0.9)$	8	8	1	2
C3	$5 - 5(t - 0.9)$	$8 - 30(t - 0.9)$	2	$2 + 5(t - 0.9)$	4	2	2	3
	Cycle 5 (batches 121–150)							
	$t \in [1.2, 1.4)$ (batches 121–140)				$t \in [1.4, 1.5)$ (batches 141–150)			
C1	$5(t - 1.2)$	$5 + 15(t - 1.2)$	$4 - 10(t - 1.2)$	$1 + 10(t - 1.2)$	1	8	2	3
C2	8	$8 - 30(t - 1.2)$	$1 + 5(t - 1.2)$	2	8	2	2	2
C3	$4 - 15(t - 1.2)$	2	$2 + 5(t - 1.2)$	3	1	2	3	3

by very large classification uncertainty between batches 110 and 150. This is mainly due to the use of fixed values of parameters α and *CP* which result in a non-optimal shrinkage of the training set with elimination of useful patterns (over-shrinkage) or accumulation of incorrectly classified data (under-shrinkage).

Figure 8 shows the percentage of times in which a concept drift has been detected and, therefore, the classifier retrained, over the 50 runs of the experiment, using the SPLL method (section "Time windows–based concept drift detection methods") and the proposed concept drift detection method (section "The proposed concept drift detection method"). As expected, the proposed approach more frequently detect the drifts in correspondence of the batches characterized by a concept drift (*D*) than in correspondence of non-drifted batches (*ND*), whereas SPLL tends to be less effective in the detection. Notice that the presence of the CDD allows remarkably reducing the number of retraining of the classifier with respect to the original COMPOSE, since it does not require to update the classifier each time a new batch of data becomes available.

Figure 9 shows the evolution of the training data when the original COMPOSE and proposed method are applied. From time $t = 0.91$ to time $t = 1.21$ (columns 1 and 2 of Figure 9), although the position and shape of the three classes are modified, the data used to train the original COMPOSE and the proposed method, obtained by applying the shrinkage, are subject to a limited drift. Notice that, however, the training set of the original COMPOSE, obtained by aggregating the labeled data with the unlabeled new data, is not able to follow the evolution of the class (unsatisfactory classification performance of 46.33%), whereas the training set of the proposed method is shown able to capture the evolving condition. This is because the original COMPOSE performs the shrinkage too frequently, whereas the proposed method only when needed. Also from time $t = 1.21$ to time $t = 1.5$ (columns 2 and 3 of Figure 9), when there is a large variation of position

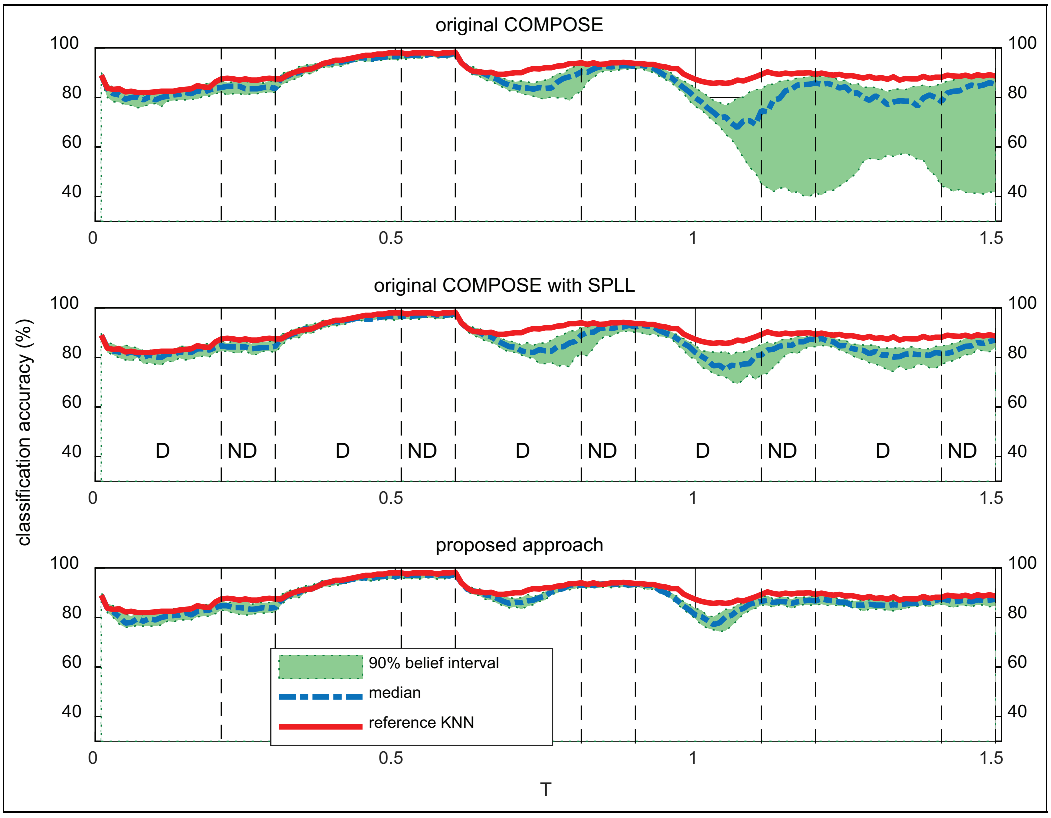

Figure 7. Performance of the four considered approaches (*D*: batches with concept drift and *ND*: batches without concept drift).

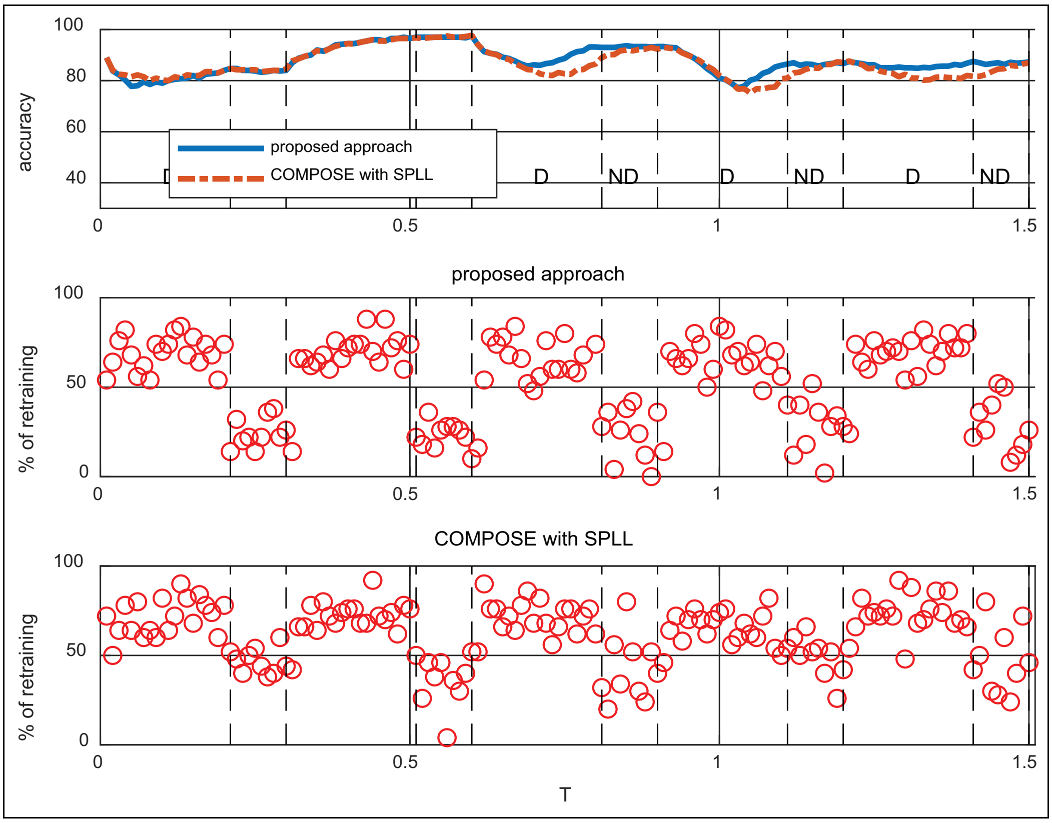

Figure 8. Percentage of times over the experiment repetition in which the classifier is retrained using the SPLL and the proposed approach.

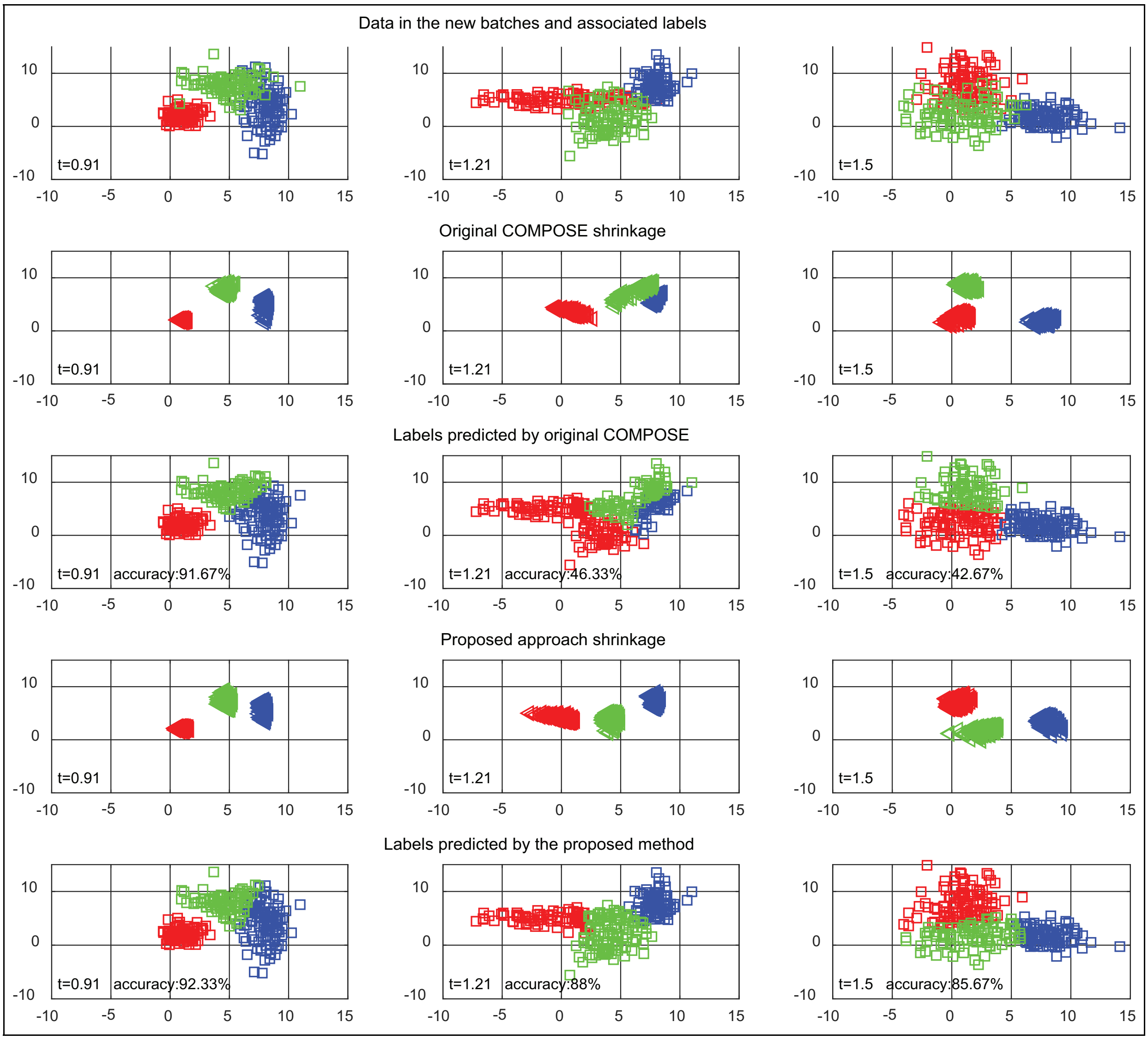

Figure 9. Evolution of the data set used to train the original COMPOSE (rows 2 and 3) and the proposed method (rows 4 and 5).

and shape of the classes, the original COMPOSE loses too much patterns and completely confuses the red and green classes, whereas the proposed method correctly follows their drifts.

Synthetic case study II

This case study is specifically designed to verify the performance of the proposed method in case of data sampled from non-Gaussian, drifting probability distributions. The two classes of data set consists of a "swing curve" (class 1) and a "moving C" (class 2).

Figure 10 shows that class 1 drifts in a clockwise rotation, whereas class 2 in a movement toward the bottom left corner. Data become available in 50 batches, each one formed by 500 patterns of both classes. Notice that the two classes overlap between batches 15 and 25 and between batches 35 and 45.

Similarly to the synthetic case study I, the experiment has been repeated 50 times, each time with a new simulation of the 50 batches of data, obtained by sampling the data from the same probability distributions.

Figure 11 shows the performance of the proposed approach and compare it with those of the other two considered approaches. Notice that when the two classes overlap (between batches 15 and 25 and between batches 35 and 45), the classification accuracy of all the approaches decreases.

The original COMPOSE with SPLL is not able to timely detect the drift. Therefore, the classifier is not updated during the phase in which the two classes overlap, which cause a problem of error accumulation and prevent a following performance recovering. Contrarily, the original COMPOSE and the proposed approach are able to follow the data distribution modifications, despite some errors in the phase in which the

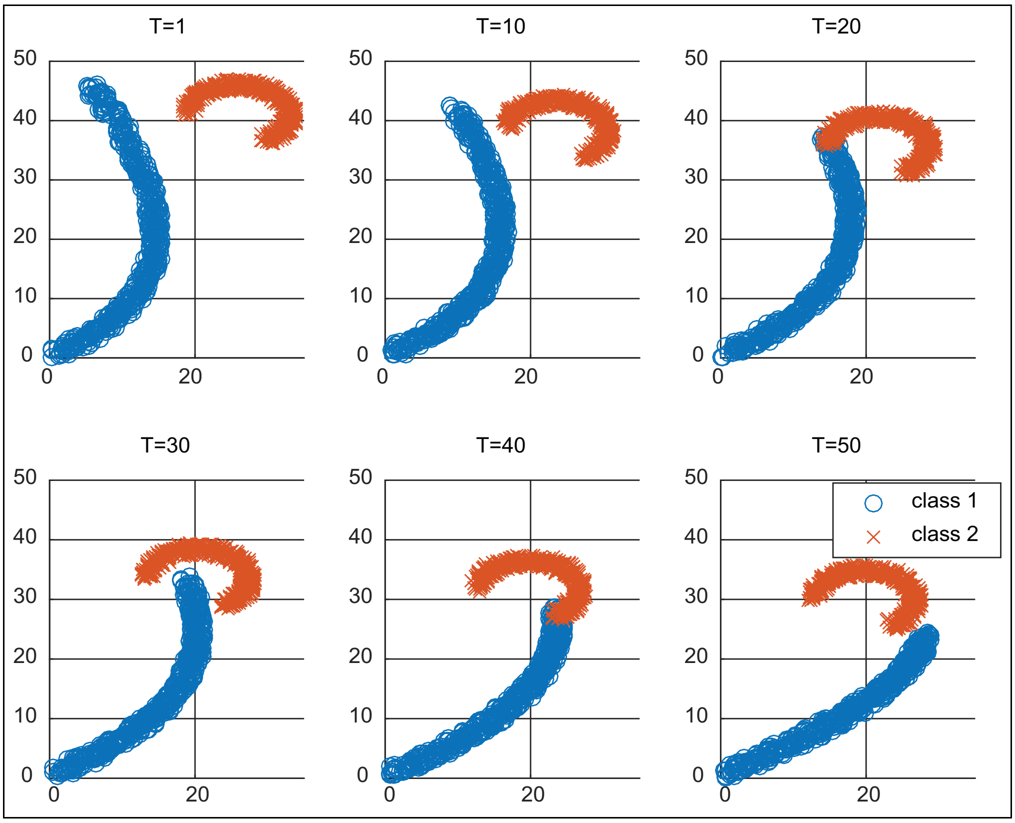

Figure 10. Sketch of irregular data stream.

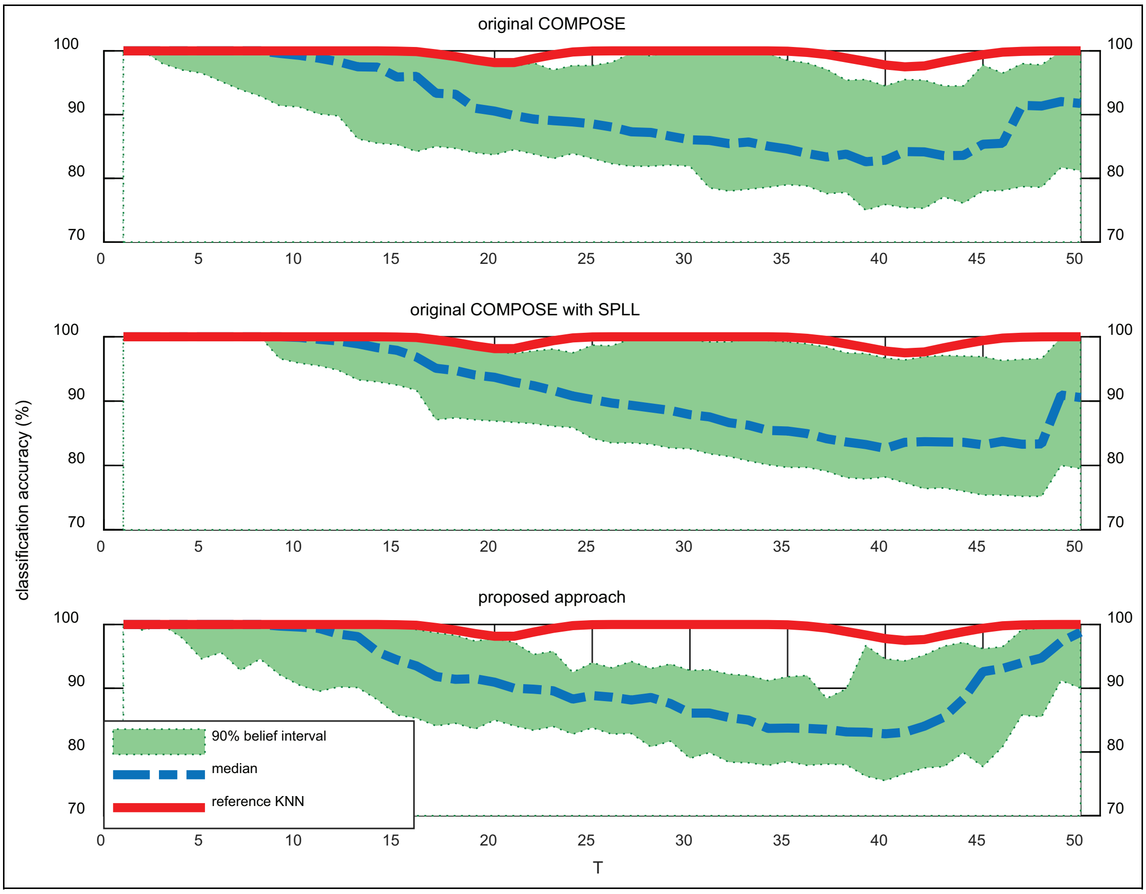

Figure 11. Performance of the four considered approaches on irregular data stream.

two classes overlap. The relatively worse performance of the original COMPOSE is due to some inappropriate updating performed between batches[8,15] and batches[40,45] and by the suboptimal setting of parameters α and CP.

Real-world case study

In this case study, we consider the diagnostic problem of classifying bearing defects using vibrational signals. Since bearings failures are one of the most frequent causes of industrial machine breakdown, they have been widely studied and several diagnostic systems have been developed for the identification of the type of defect causing the bearing degradation.[46,47] In this work, we focus on the classification of bearing defects in industrial machines characterized by variable speed and load profiles, such as those used in automotive powertrains. The development of the diagnostic method is complicated by the impossibility of training the diagnostic model using data describing all the possible combinations of operational conditions that can be encountered during operation.

This case study is designed using data taken from the Case Western Reserve University bearing data set,[28] which contains vibrational signals measured by three accelerometers during 72 laboratory tests performed on ball bearings located in two different positions of a powertrain (fan and drive end).[28,47] The laboratory tests have been performed considering three different bearing degradation modes (inner race, outer race and ball defects), three different degradation levels (7, 14 and 21 mil inches defects) and four different loads (0, 1, 2 and 3 horsepowers) applied to the powertrain.

The raw vibrational signals obtained in each experimental test have been divided into 10 time windows. From each time window, 87 different features, such as statistical indicators and time-frequency transform coefficients, have been extracted according to the approach illustrated in Baraldi et al.[48] Thus, a 720×87-dimensional data set is available, corresponding to 60 patterns for each bearing defect class and motor load. Since it has been shown that irrelevant and noisy features unnecessarily increase the complexity of the classification problem and can degrade modeling performance,[49] we have applied the feature selection wrapper approach described in Baraldi et al.[48] to select three features to be used as input to the diagnostic system. Notice that the feature selection approach is here applied considering only 180 patterns extracted from the 0 horsepowers load data. Table 2 reports the three selected features.

Table 2. Selected features.

	Feature name
1	Kurtosis
2	Normalized symlets 10 wavelet coefficient
3	Maximum Haar wavelet coefficient

To verify the capability of the proposed diagnostic method in an evolving environment, we have designed two different experiments, characterized by a different order in which the batch of data becomes available. In both experiments, we assume that the first batch of data is formed by all the 180 patterns collected when the powertrain is operating at a load of 0 horsepowers and that the labels of these patterns are available. This first batch of data is used to train the diagnostic model. The remaining 18 batches are formed by 30 patterns which are all collected when the motor is working at the same load, that is, always at 1, 2 or 3 horsepowers.

In the first experiment, we consider a gradual modification of the powertrain load, obtained by assuming that the sequence of the loads in the batches is [1, 2, 3, 3, 2, 1, 1, 2, 3, ...], whereas in the second test, we randomly sample the batch sequence, allowing irregular modifications of the powertrain loads, with possible jumps between nonconsecutive load levels. The two experiments have been repeated 100 times, each time considering the same sequence of loads but a different random composition of the data batches.

We have considered two possible classification algorithms for developing the FC: the KNN[45] and the support vector machine (SVM).[50,51] Given the more satisfactory performance of SVM on the load 0 horsepowers data, the diagnostic model is built using SVM as classification algorithm.

The performance of the proposed method in the two experiments is compared with that of (1) a pure SVM classifier trained using the initial 180 labeled patterns and never updated and (2) the original COMPOSE with and without SPLL.

Figures 12 and 13 show the obtained average performances over 100 test repetitions. Notice that

- The proposed method is more accurate than the other three methods, especially as time passes and the number of times that the model is retrained increases. This is due to the effect of the CDD.
- The proposed method retrains the classifier less time than COMPOSE with SPLL, in both cases of regular load and random load experiments (bottom of Figures 12 and 13).
- The performance of the pure SVM is, on average, the less satisfactory than that of the COMPOSE-based methods, since it does not consider the modification of the load. In few cases, in which the COMPOSE with and without SPLL apply inappropriate shrinkages that lead to the elimination of useful data from the training set, the pure SVM is more accurate. On the other side, the proposed method is shown able to mitigate this negative effect of the other two COMPOSE methods.
- The proposed method is less accurate than the original COMPOSE and the COMPOSE with SPLL in the classification of few batches (batch 3 in Figure

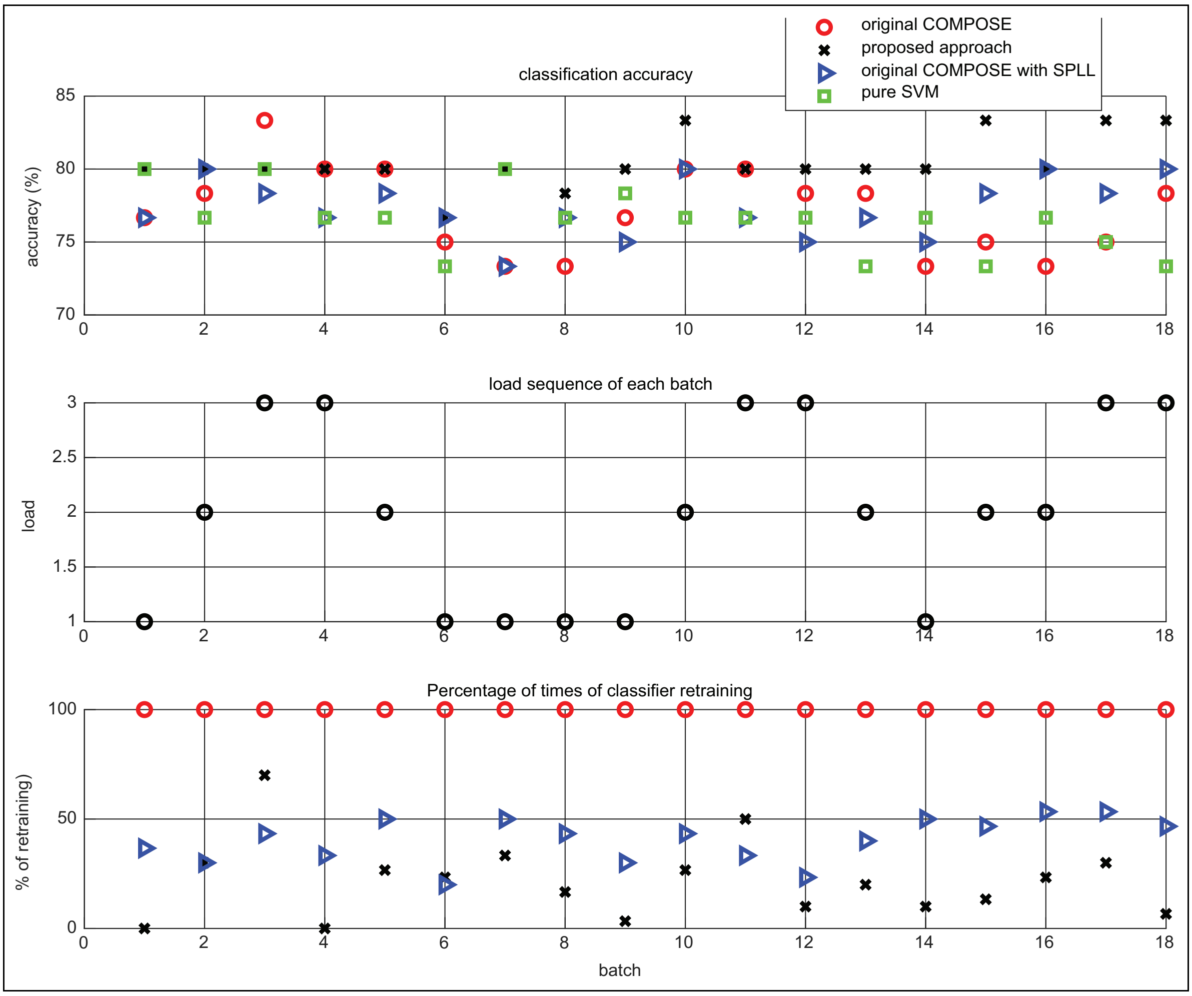

Figure 12. Average accuracy (top), load sequence (middle) and model retraining percentage (bottom) in the experiment with regular load.

Table 3. Average classification accuracy of the proposed method and two other approaches considering different numbers of patterns per batch.

	Number of pattern in a batch	Original COMPOSE	Original COMPOSE with SPLL	Proposed method
Regular load modifications	10	68.25	73.56	76.59
	30	68.48	75.21	77.40
	60	78.70	78.98	79.19
	90	79.13	79.15	80.20
	180	79.52	79.74	80.57
Random load modifications	10	68.13	72.79	76.12
	30	68.25	75.01	77.23
	60	77.56	77.98	78.16
	90	77.93	77.95	78.02
	180	77.77	77.96	78.01

COMPOSE: compacted object sample extraction; SPLL: semi-parametric log-likelihood.

12 for the original COMPOSE and batch 16 in Figure 13 for the COMPOSE with SPLL). This is because the classification model has not been updated due to missed detections of drifts by the α shape–based method.

Finally, it is interesting to observe that in Baraldi et al.[52] the authors have considered the same data set and trained the classifier using 720 labeled patterns taken from all the four load levels and obtained an average accuracy of 79.78%, which is comparable to

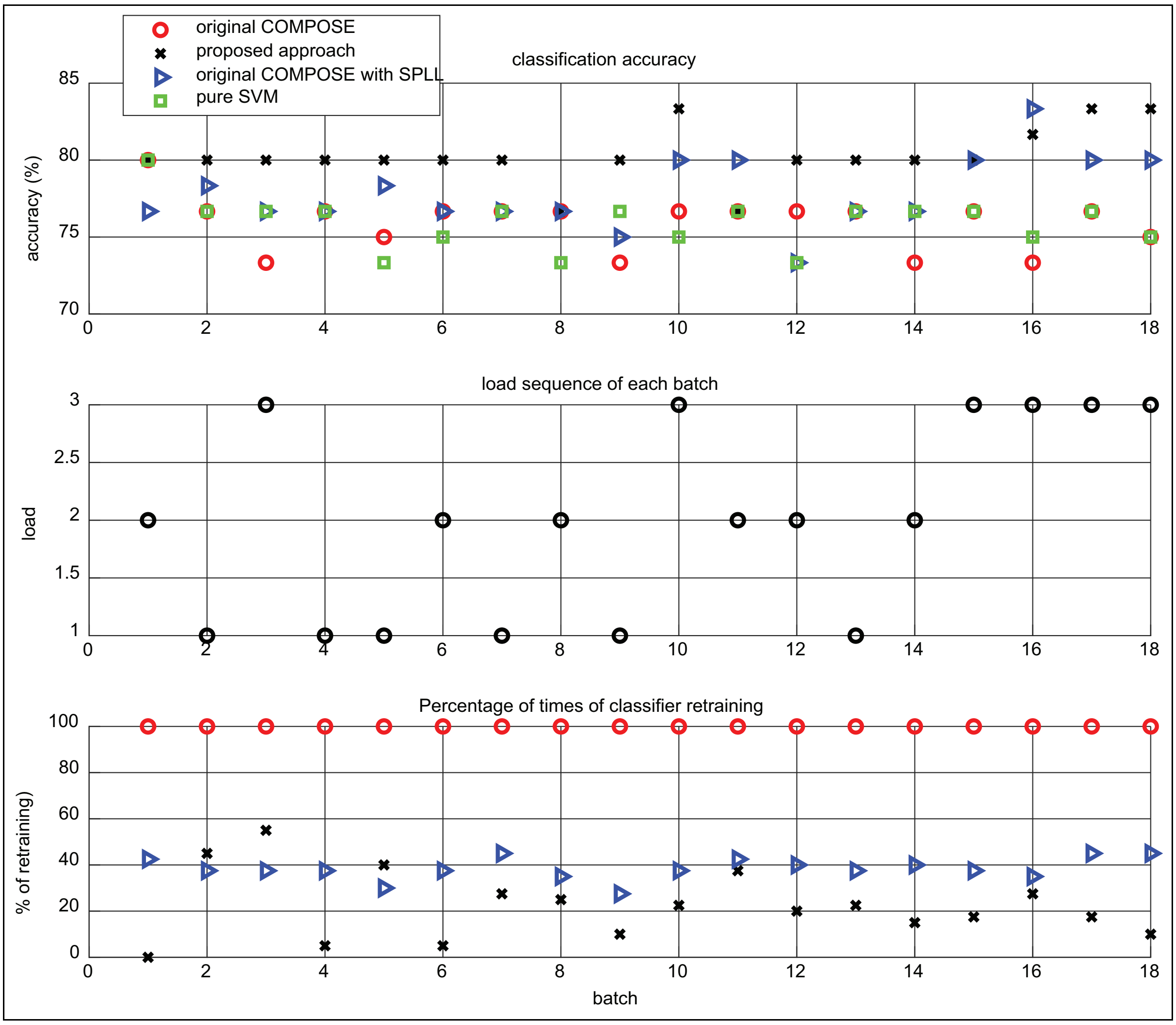

Figure 13. Average accuracy (top), load sequence (middle) and model retraining percentage (bottom) in the experiment with random load.

the average accuracy of the proposed method (Table 3, last column). This shows that the proposed method is able to achieve a similar accuracy, but in the more challenging situation in which only labeled data from load 0 are available.

The two experiments have also been performed varying the number of patterns per batch. Table 3 reports the obtained average classification accuracy. Notice that

1. The performance of all the methods tends to increase as the number of patterns per batch becomes larger. The reason is that larger batches of data contain more information and facilitate the identification of the overlapping regions between the old and new data.
2. The proposed approach is less sensitive to the number of patterns per batch. This is because it updates the classifier only when the concept drift is detected in the batch, avoiding unnecessary retraining with small batches of data that reduces the performance.

Conclusion

This article proposes a novel method to perform fault diagnostics in an evolving environment characterized by continuous or periodic modifications of the working conditions. The proposed approach, which is inspired by the COMPOSE algorithm, allows correctly diagnosing faults in different operational conditions. Furthermore, it allows retraining the classification model when new batches of data become available, without the necessity of knowing the class of the new data. The approach is shown to outperform COMPOSE from the point of view of classification accuracy, parsimony in the number of model retraining and easiness of parameter settings.

The method is based on the hypothesis that the concept drift caused by the evolving environment is gradual, rather than abrupt: future work will be devoted to investigating techniques for the verification of the gradual drift hypothesis and of its consequences. It also seems interesting to address the problem of selecting the features to be used for fault classification in an evolving environment, that is, how to dynamically identify a suitable set of features for fault diagnostics, taking into account that the available labeled data do not refer to all the possible working conditions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by China NSFC under Grant No. 61703431.

ORCID iDs

Yang Hu https://orcid.org/0000-0002-6255-6376
Jie Liu https://orcid.org/0000-0003-0895-7598
Enrico Zio https://orcid.org/0000-0002-7108-637X

References

1. Xia T, Dong Y, Xiao L, et al. Recent advances in prognostics and health management for advanced manufacturing paradigms. *Reliab Eng Syst Saf* 2018; 178: 255–268.
2. Chang S, Gao L and Wang Y. A review of integrated vehicle health management and prognostics and health management standards. In: *Proceedings of the international conference on sensing, diagnostics, prognostics, and control (SDPC)*, Xi'an, China, 15–17 August 2018, pp.476–481. New York: IEEE.
3. Dong Y, Xia T, Fang X, et al. Prognostic and health management for adaptive manufacturing systems with online sensors and flexible structures. *Comput Ind Eng* 2019; 133: 57–68.
4. Tinga T and Loendersloot R. Aligning PHM, SHM and CBM by understanding the physical system failure behaviour. In: *Proceedings of European conference of the prognostic and health management society*, Nantes, 17 July 2014. Nantes: PHM Society.
5. Lee J, Wu F, Zhao W, et al. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mech Syst Signal Pr* 2014; 42: 314–334.
6. Zio E. Prognostics and health management of industrial equipment. In: S Kadry (ed.) *Diagnostics and prognostics of engineering systems: methods and techniques*. Hershey, PA: IGI Global, 2012, pp.333–356.
7. Lei Y, Yang B, Jiang X, et al. Applications of machine learning to machine fault diagnosis: a review and roadmap. *Mech Syst Signal Pr* 2020; 138: 106587.
8. Zhao F, Tian Z, Bechhoefer E, et al. An integrated prognostics method under time-varying operating conditions. *IEEE T Reliab* 2015; 64(2): 673–686.
9. Bian L and Gebraeel N. Stochastic methodology for prognostics under continuously varying environmental profiles. *Stat Anal Data Min* 2013; 6(3): 260–270.
10. Razavi-Far R, Baraldi P and Zio E. Dynamic weighting ensembles for incremental learning and diagnosing new concept class faults in nuclear power systems. *IEEE T Nucl Sci* 2012; 59(5): 2520–2530.
11. Kolter JZ and Maloof MA. Dynamic weighted majority: an ensemble method for drifting concepts. *J Mach Learn Res* 2007; 8: 2755–2790.
12. Chen H, Tino P, Rodan A, et al. Learning in the model space for cognitive fault diagnosis. *IEEE T Neur Net Lear* 2014; 25(1): 124–136.
13. Zliobaite I, Bifet A, Pfahringer B, et al. Active learning with drifting streaming data. *IEEE T Neur Net Lear* 2014; 25(1): 27–39.
14. Ben-David S and Blitzer J. Analysis of representations for domain adaptation. *Adv Neural Inf Process Syst* 2007; 19: 137–144.
15. Daume H III and Marcu D. Domain adaptation for statistical classifiers. *J Artif Intell Res* 2006; 26: 101–126.
16. Dong M and He D. A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology. *Mech Syst Signal Pr* 2007; 21(5): 2248–2266.
17. Li J, Zhang X, Zhou X, et al. Reliability assessment of wind turbine bearing based on the degradation-Hidden-Markov model. *Renew Energy* 2019; 132: 1076–1087.
18. Li F and Xu J. A new prognostics method for state of health estimation of lithium-ion batteries based on a mixture of Gaussian process models and particle filter. *Microelectron Reliab* 2015; 55(7): 1035–1045.
19. Karami M and Wang L. Fault detection and diagnosis for nonlinear systems: a new adaptive Gaussian mixture modeling approach. *Energy Build* 2018; 166: 477–488.
20. Dyer KB, Capo R and Polikar R. COMPOSE: a semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE T Neur Net Lear* 2014; 25(1): 12–26.
21. Dyer KB and Polikar R. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. In: *Proceedings of the international joint conference on neural networks (IJCNN)*, Brisbane, QLD, Australia, 10–15 June 2012, pp.1–9. New York: IEEE.
22. Bose RPJC, van der Aalst WMP, Zliobaite I, et al. Dealing with concept drifts in process mining. *IEEE T Neur Net Lear* 2014; 25(1): 154–171.
23. Polikar R and Alippi C. Guest editorial learning in nonstationary and evolving environments. *IEEE T Neur Net Lear* 2014; 25(1): 9–11.
24. Hu Y, Baraldi P, Di Maio F, et al. A compacted object sample extraction (COMPOSE)-based method for fault diagnostics in evolving environment. In: *Proceedings of the prognostics and system health management conference (PHM)*, Beijing, China, 21–23 October 2015, pp.1–5. New York: IEEE.
25. Guo B, Menon J and Willette B. Surface reconstruction using alpha shapes. *Comput Graph Forum* 1997; 16: 177–190.
26. Xu X and Harada K. Automatic surface reconstruction with alpha-shape method. *Vis Comput* 2003; 19: 431–443.
27. Case Western Reserve University Bearing Data Center Website, 2015, https://github.com/Litchiware/cwru
28. Korbicz J. *Fault diagnosis: models, artificial intelligence, applications*. Washington, DC: U.S. Government Printing Office, 2004.

29. Samanta B and Al-Balushi KR. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mech Syst Signal Pr* 2003; 17: 317–328.
30. Kuncheva LI and Faithfull WJ. PCA feature extraction for change detection in multidimensional unlabeled data. *IEEE T Neur Net Lear* 2014; 25(1): 69–80.
31. Venkatasubramanian V. A review of process fault detection and diagnosis: part III: process history based methods. *Comput Chem* 2003; 27: 293–311.
32. Lei Y, Lin J, He Z, et al. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mech Syst Signal Pr* 2013; 35(1–2): 108–126.
33. Lemos A, Caminhas W and Gomide F. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Inf Sci* 2013; 220: 64–85.
34. Sakthivel NR, Sugumaran V and Nair BB. Automatic rule learning using roughset for fuzzy classifier in fault categorization of mono-block centrifugal pump. *Appl Soft Comput* 2012; 12(1): 196–203.
35. Rokach L and Hutter D. Automatic discovery of the root causes for quality drift in high dimensionality manufacturing processes. *J Intell Manuf* 2011; 23(5): 1915–1930.
36. Horstmann J, Spahr H, Buj C, et al. Full-field speckle interferometry for non-contact photoacoustic tomography. *Phys Med Biol* 2015; 60(10): 4045–4058.
37. C C Aggarwal. *Data streams: models and algorithms*, vol. 31. Cham: Springer, 2007.
38. Gonçalves PM, de Carvalho Santos SGT, Barros RSM, et al. A comparative study on concept drift detectors. *Expert Syst Appl* 2014; 41(18): 8144–8156.
39. Kuncheva LI. Change detection in streaming multivariate data using likelihood detectors. *IEEE T Knowl Data En* 2013; 25: 1175–1180.
40. Dries A and Rückert U. Adaptive concept drift detection. *Stat Anal Data Min* 2009; 2(5–6): 311–327.
41. Zweig MH and Campbell G. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clin Chem* 1993; 39(4): 561–577.
42. Edelsbrunner H and Mucke EP. 3-dimensional alpha-shapes. *ACM T Graphic* 1994; 13: 43–72.
43. Kim DS, Seo J, Kim D, et al. Three-dimensional beta shapes. *CAD Comput Aided Des* 2006; 38: 1179–1191.
44. Lei Y and Zuo MJ. Gear crack level identification based on weighted K nearest neighbor classification algorithm. *Mech Syst Signal Pr* 2009; 23(5): 1535–1547.
45. Xu Z, Xuan J, Shi T, et al. A novel fault diagnosis method of bearing based on improved fuzzy ARTMAP and modified distance discriminant technique. *Expert Syst Appl* 2009; 36(9): 11801–11807.
46. Wu S-D, Wu P-H, Wu C-W, et al. Bearing fault diagnosis based on multiscale permutation entropy and support vector machine. *Entropy* 2012; 14(8): 1343–1356.
47. Baraldi P, Cannarile F, Di Maio F, et al. Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable. *Reliab Eng Syst Saf* 2015; 56: 1–13.
48. Emmanouilidis C, Hunter A, MacIntyre J, et al. Selecting features in neurofuzzy modelling by multiobjective genetic algorithms. In: *Proceedings of the ninth international conference on artificial neural networks, ICANN 99 (Conf. Publ. No. 470)*, Edinburgh, 7–10 September 1999, vol. 2, pp.749–754. New York: IEEE.
49. Langone R, Alzate C, De Ketelaere B, et al. LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines. *Eng Appl Artif Intell* 2015; 37: 268–278.
50. Chang C and Lin C. LIBSVM: a library for support vector machines. *ACM T Intel Syst Tec* 2001; 2: 1–30.
51. Baraldi P, Cannarile F, Di Maio F, et al. Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable conditions. *Eng Appl Artif Intell* 2016; 56: 1–13.

Appendix 1

Figure 14 reports the overall procedure used for the diagnostics in an evolving environment, which takes into account both the concept drift detection and the retraining of FC by updating the training set.

The details of the procedure are as follows:

1. Collect the initial training set $T = \{x_T^i, l_T^i\}, i = 1, 2, \ldots, n_T, l_T = 1, 2, \ldots, n_l$, to train an empirical classifier; n_l is the total number of the classes.
2. As in step (a) of the original COMPOSE, collect the unlabeled coming set $C = \{x_C^j\}, j = 1, 2, \ldots, n_C$ and use the empirical classifier to label all the patterns in C to output $C = \{x_C^j, l_C^j\}$.
3. For class k ($k = 1, 2, \ldots, n_l$), perform the α shape–based drift detection method (mentioned in subsection "The proposed concept drift detection method") on all the patterns in T whose label are k, namely, $T_k = \{\mathbf{X}_T | l_T = k\}$, are the training pattern set, and all the patterns in C whose label are k, namely, $C_k = \{\mathbf{X}_C | l_C = k\}$, are the test pattern set; output the drifted set D_k and the non-drifted pattern set ND_k, count the number of patterns in T_k and D_k and marked them as n_{T_k} and n_{D_k}, respectively.
4. If $n_{D_k} = 0$, go to step 5, otherwise, perform COMPOSE by the following steps:
   - Build up the aggregated pattern set $A_k = T_k \cup D_k$, count the number of patterns in A_k and mark it as $n_{\mathrm{A}_k\_ini}$.
   - Calculate the α value of A_k using equation (2) and set the value of CP as
   $$CP = 1 - \frac{n_{\mathrm{T}_k}}{n_{\mathrm{A}_k\_ini}}$$
   - Perform the shrinkage on A_k using α and CP, output the shrunk set $A_k^{'}$.
   - Set $D = \varnothing$, go to step 6.
5. Keep the training set T unchanged, set $k = k + 1$ and go back to step 4 until all the classes are searched; then, go to step 7.

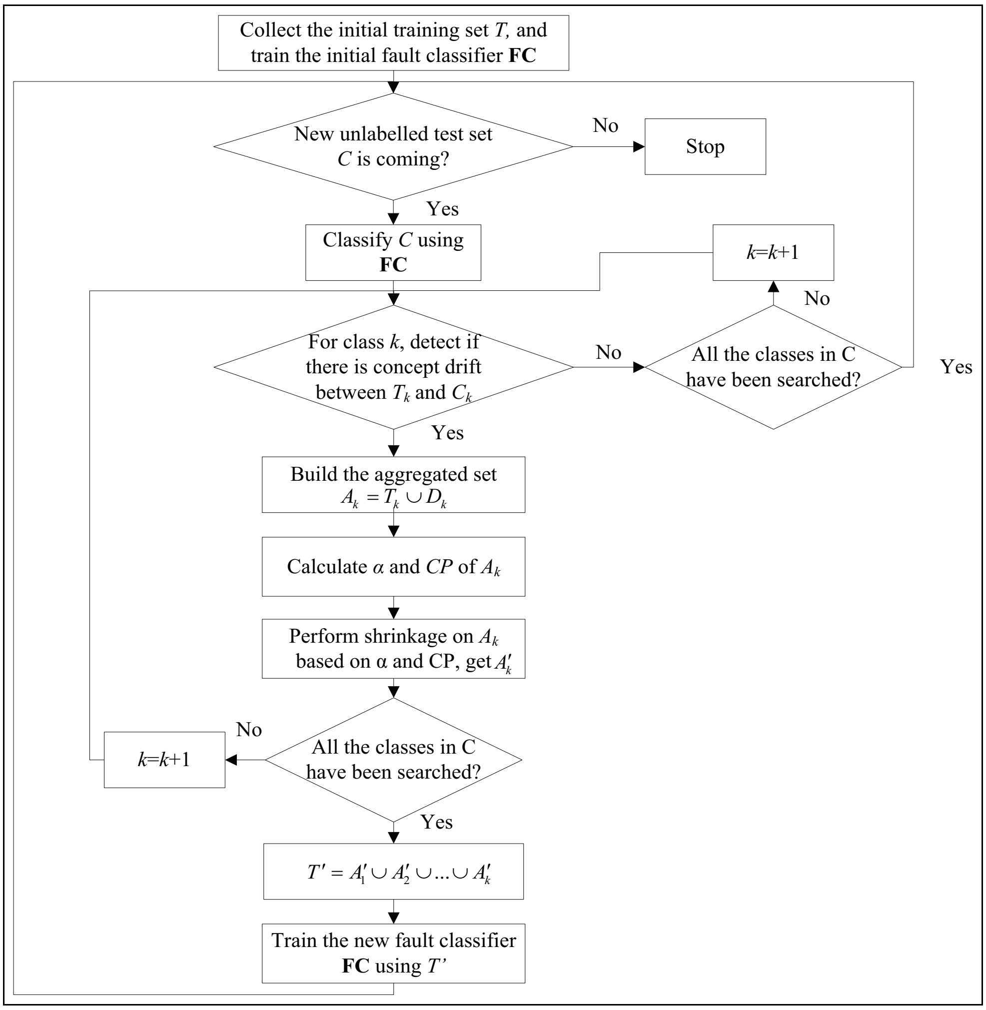

Figure 14. Flowchart of the proposed approach.

6. Set $k = k + 1$ and go back to step 4 until all the classes are searched.
7. Update the new training set $T' = A'_1 \cup A'_2 \cup ... \cup A'_k$ and retrain the FC using T'.
8. If new patterns are coming, go back to step 2, otherwise stop the algorithm.