Fault Diagnostics in an Evolving Environment by COMPacted Object Sample Extraction (COMPOSE) algorithm

Yang Hu[1], Piero Baraldi[1], Francesco Di Maio[1], Enrico Zio[1,2]

[1] *Politecnico di Milano, Department of Energy,Milan, Italy*

[2] *Chair on Systems Science and the Energetic challenge, Fondation EDF, Centrale Supélec, Paris, France*

ABSTRACT

Through the course of their life, industrial components and systems operate in an evolving environment characterized by continuous or periodic modifications of the working conditions. Diagnostic methods may suffer such an evolving environment if the data used to develop them are collected in a limited set of working conditions, not sufficiently represent the possible environments that may be experienced during life. Thus, in order to overtake this problem, we propose a novel approach based on the COMPacted Object Sample Extraction (COMPOSE) algorithm, which has been previously developed in the domain of streaming data learning and is here adapted to deal with the task of fault diagnostics. The developed diagnostic method is shown able to correctly classify data taken from synthetic and real-world case studies.

Key words: Concept Drift, Drift Detection, Fault Diagnostics, α Shape Reconstruct, Evolving Environment, Bearing Faults

1. INTRODUCTION

Fault diagnostics is an essential step for prognostics and health management of components and systems [1,2]. In the last decades, a number of methods based on various recognition techniques such as K Nearest Neighbor (KNN) [3], Artificial Neural Network (ANN) [4,5], Support Vector Machine (SVM) [6], Relevance Vector Machine (RVM) [7], have been developed for the classification of the type of fault causing the malfunctioning of components and systems. The proposed methods have been reported to achieve satisfactory performance in several works [3–5,7,8]. 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 different from those considered to train the model, its performance maybe unsatisfied. Such problem occurs for components like bearings, gears, alternators, shafts, pumps, etc. operating in an evolving environment characterized by modifications of the working conditions, e.g., applied loads [9,10].

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 [11–14]. For example, in [11], the authors propose an ensemble approach where a new model is developed and added to the ensemble each time a new set of data becomes available. In [15,16], the problem has 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.

In this work, we present the development of a systematic fault diagnostics approach with continuous and automatical updating of the diagnostics model without the requirement of 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 [17,18]. COMPOSE allows obtaining satisfactory classification performances in presence of concept drifts, i.e. situations in which the statistical properties of the class of the patterns change over time in an unforeseen way [19–21]. It has been applied with success to the classification of several published artificial testing datasets and a real-world weather dataset [17]. 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 dataset 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. 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 when 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 the present paper is the development of a new approach for concept drift detection. The method, based on the construction of **α** shape surfaces [22], allows identifying when it is really necessary to update the classification model due to the presence of a concept drift. 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 method is verified with respect to two datasets containing concept drifts: 1) a synthetic dataset made by artificial unimodal Gaussian data, inspired from the case study named "Unimodal Gaussian With Added Class" in [17] and a real-world bearing vibration dataset taken from Case Western Reserve University [23].

The remainder of this paper is organized as follows: Section 2 and 3 illustrate the problem of fault diagnostics in an evolving environment; Section 4 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 5; Section 6 shows the performance of the proposed approach on the two case studies; Section 7 summarizes the whole paper and puts forward some future work for improvement of the proposed method.

2. PROBLEM STATEMENT

Fault diagnostics is typically considered a supervised classification problem [2,24]. Signal measurements are given in input to an empirical classifier, which provides in output 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 couples: 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, i.e. 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, 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 [25], 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.

3. 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 fault classifier once a concept drift has been detected.

Let us assume that a fault classifier *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 fault classifier, i.e. a tentative class label $\widetilde{l_i^C}$ is assigned to each pattern. Then, *C* is sent to the concept drift detector (CDD): in case of no concept drift detection the classification labels are confirmed, whereas, in case of concept drift detection, the fault classifier has to be updated. This requires the construction of a new training set, which will be used for retraining the fault classifier (*FC*). Finally, the updated classifier can be used for the classification of the patterns in C.

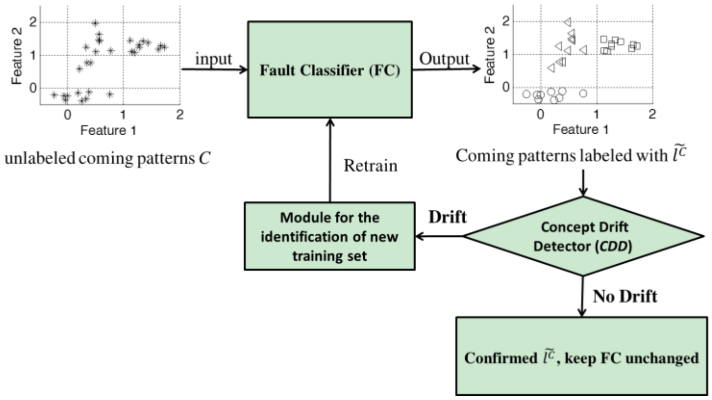

Figure 1 Sketch of overall fault diagnostic method

In this work, we focus our attention on the concept drift detector (Section 4) and on the method for the construction of the training set which is used for retraining the fault classifier (Section 5), whereas the discussion of classification algorithms for fault diagnostics is outside the scope of the present work (The interested reader can refer to [26–29] for examples of application of empirical classification algorithms to fault diagnostic problems).

4. CONCEPT DRIFT DETECTION

The problem of concept drift detection has been extensively studied for its potential applications in very different areas such as quality control [30], medical condition monitoring [31] and market analysis [32,33]. In the framework of fault diagnosis, 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.

4.1. 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 [25,34] divergence and the Hotelling's T-squared distribution [35] are typically employed for this aim. In [25], the authors have shown the effectiveness of an approach based on the use of the Semi-Parametric Log-Likelihood (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:

$$SPLL(W_1, W_2) = \frac{\mathbf{1}}{M_2} \sum_{\boldsymbol{x} \in W_2} \max_{i=\mathbf{1}}^{N} \left(P\left(\boldsymbol{x} | W_1 \right) \right) \tag{4.1}$$

where W_1 and W_2 are 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 can be decided by observing the Receiver Operating Characteristic (ROC) curve [36] and choosing the threshold value which allows achieving the best tradeoff between missed and false detections [25].

The SPLL method tends to fail when it is applied to data characterized by very irregular probability distributions, such as that 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 Section 3.1.

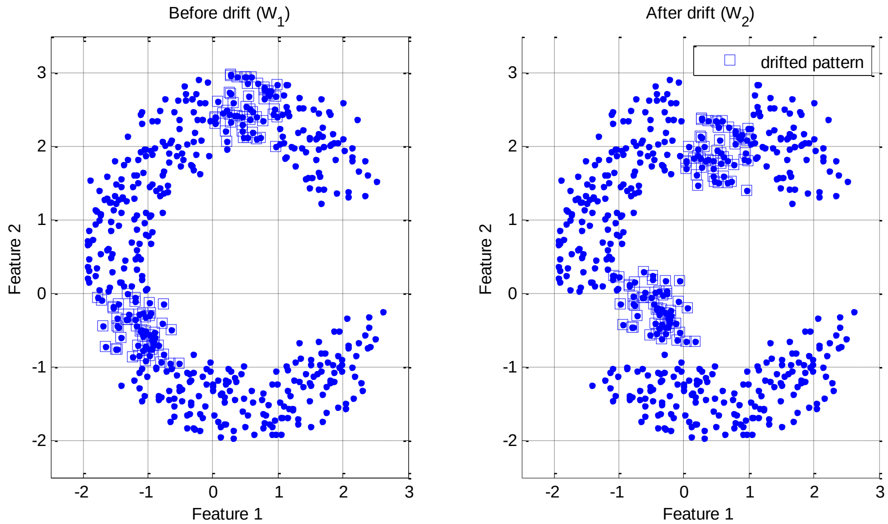

Figure 2 Data distributions before (left) and after (right) the concept drift occurrence.

4.2. The proposed concept drift detection method

The proposed method for concept drift detection is based on the use of the **α** shapes reconstruction technique [22], which allows obtaining a surface enveloping a dense unorganized set of multidimensional patterns. A detailed description of the method can be found in [22,37,38]. Here, we focus on the choice of the parameter α, which controls the level of details and roughness of the enveloping surface. According to [37], α can be seen as the radius of a hyper-spherical spoon, which is used to 'carve out' around the dataset.

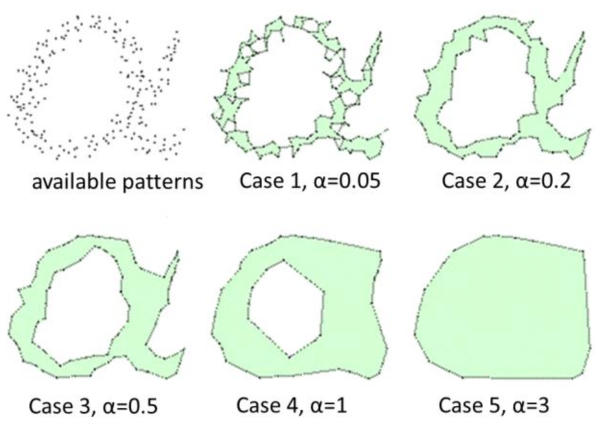

Figure 3. Top right: avavilable patterns, Case 1-5: α shape reconstruction surfaces obtained for different values of the parameter α (from [37]**)**

Figure 3 shows the enveloping surfaces obtained by applying the α shapes 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 [1], 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 dataset.

According to the analysis in [22,37], the value of the parameter α should be set considering the density of the data patterns: higher the density, smaller the α value. In this work, α is set considering the average minimum distance among the patterns:

$$\alpha = \frac{1}{n}\sum_{i}^{n}\left(\min_{j=1,\, j\neq i}^{n}\left(d_{ij}\right)\right) \tag{4.2}$$

where d_{ij} is the Euclidean distance between patterns $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, 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 $T = \{\boldsymbol{x}_i^T\}, i = 1,2, \dots, n_T$, the following procedure is proposed for the detection of a concept drift in a new batch of test patterns, $C = \{\boldsymbol{x}_j^C\}, (j = 1,2, \dots, n_C)$

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 (4.2) to the training set, T, patterns;
4) Build the α shape surface S_T of the training set and compute its iper-volume V_T;
5) Set the indicator of the pattern in the test batch, j, equal to 1;
6) Build a new dataset, A_j, as the union of the training set T and the j-th pattern of the test set batch, C, i.e. $\mathrm{A}_j = \{T \cup \boldsymbol{x}_j^C\}$;
7) Build the α shape surface S_{A_j} and compute its volume V_{A_j};
8) Find the ratio $R_j = V_{\mathrm{A}_j}/V_T$; if $R_j > f$, put $\boldsymbol{x}_j^C$ into the subset of the drifted patterns, i.e. $D = D \cup \boldsymbol{x}_j^C$, otherwise, put $\boldsymbol{x}_j^C$ into the subset of the non-drifted patterns, i.e. $ND = ND \cup \boldsymbol{x}_j^C$;
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 outputs the two subset of the drifted, D, and non-drifted, ND, patterns.

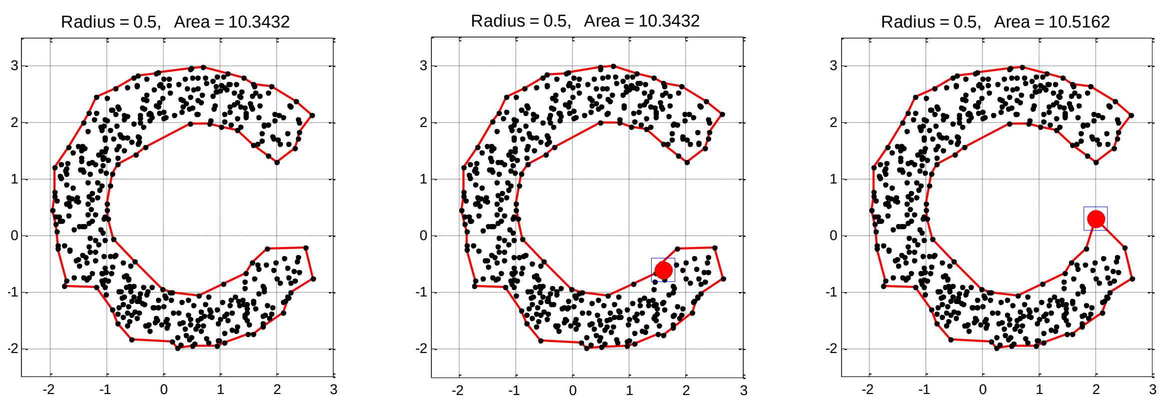

Figure 4 Sketch of α shape based drift detection method

Figure 4 shows an example of application of the α shape based drift detection method. 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 leave unmodified the area of the α-shape surface; whereas 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 of the area.

5. CONSTRUCTION OF THE NEW TRAINING SET

According to the scheme of Figure 1, once a concept drift has been detected, the diagnostics model has to be updated. i.e. a new training set has to be constructed and used to develop 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 [17], this assumption is met when the concept drift is not abrupt but gradual, i.e. the training set T and the test set C must have some overlapping area, as shown in Figure 5 (left). 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 on system, or to seasonal variations in the external environment.

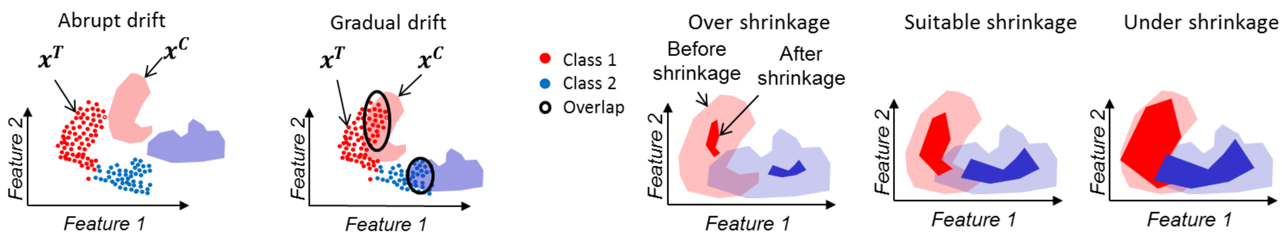

Figure 5 Sketch of abrupt & gradual drift (left) and degree of shrinkage (right)

Assuming that a fault classifier has been already trained using the labeled data in T and that a concept drift has been identified in the test set C, the method that we propose for the construction of a new training set is based on the following steps:

a) an aggregate dataset formed by the labeled patterns of the training set T and those of C with corresponding labels assigned by the empirical fault classifier is built;

b) the α shape surface reconstruction is applied to find the surface boundary of each class in the aggregated dataset; parameter α allows to regulate the level of details and roughness of the surfaces;

c) the core regions of each class are identified by applying a proper shrinkage to the obtained class surface boundaries; the shrinkage is achieved by removing the patterns which are on the surface of the α shape: parameter *CP* is used to regulate the desired shrinkage percentage (the details of the shrinkage procedures can be found in [17]);.

d) the new training set is formed by all the labelled patterns in the core regions identified in c).

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 procedure proposed in this work differs from that of COMPOSE since:

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) a novel dynamic procedure for setting the parameters α and *CP* is proposed.

With respect to the setting of α, its impact on the obtained surfaces has been already illustrated in Section 4.1 and the α parameter can be set according to equation (4.2). With respect to *CP*, notice that if too large values are used, an over-shrinkage of the α-surfaces can arise, with consequent elimination of useful patterns from the training set found in b), and, thus, inability of updating the algorithm in order to follow the concept drift. On the other side, too small *CP* values result in too large core regions (under-shrinkage) with consequent overlapping of core regions of different classes (Figure 5, right). This can deteriorate the performance of the classifier since it causes the inclusion of patterns with wrong label in the new training set.

In order to obtain a satisfactory shrinkage, the value of CP is set as

$$CP = 1 - \frac{n_{\mathrm{T}}}{n_{\mathrm{A}}} \tag{5.1}$$

where n_T is the number of the patterns in the training dataset, n_A is the number of the patterns in the aggregated dataset. The main advantage of this procedure for setting the parameter *CP* is that it maintains the number of patterns in T unchanged after shrinkage, avoiding the phenomena of under and over-shrinkage. Notice that this strategy, differently from that in [17], allows using different *CP* values for different classes of patterns.

The details of the execution procedure and the flow chart of the proposed approach are given in Appendix 2.

6. CASE STUDIES

In this Section, we verify the proposed method considering a synthetic (Section 6.1) and a real-world (Section 6.2) case study. The synthetic dataset, taken from [17], is characterized by patterns sampled from bi-dimensional Gaussian distributions, whereas the real-world dataset, taken from [23], considers vibrational data collected during experimental tests on healthy and faulty bearings in different operational conditions. The performance of the proposed diagnostic approach has been compared with that obtained by applying the original COMPOSE algorithm, with parameters α and *CP* set to 0.6, according to the suggestions in [17], and updating the diagnostic model each time a new batch of data becomes available.

6.1. The synthetic case study

In this case study, data 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 means and standard deviations 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 5 cycles. In each cycle, there are 20 consecutive batches in which the data distributions are slowly changing (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 an component or system characterized by phases with changing load followed by phases with constant load.

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)							
	$t\epsilon$[0,0.2), (batches 1 - 20)				$t\epsilon$[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)							
	$t\epsilon$[0.3,0.5), (batches 31 - 50)				$t\epsilon$[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)							
	$t\epsilon$[0.6,0.8) , (batches 61 - 80)				$t\epsilon$[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)							
	$t\epsilon$[0.9,1.1) , (batches 91 - 110)				$t\epsilon$[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)	6	2	2	3
	Cycle 5 (batches 121 - 150)							
	$t\epsilon$[1.2,1.4) , (batches 121 - 140)				$t\epsilon$[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	6-25(t-1.2)	2	2+5(t-1.2)	3	1	2	3	3

6.1.1. Results

The KNN algorithm [3] has been used as base classifier and the experiment has been repeated 50 times, each time with a new simulation of the batches of data, using the distributions of Table 1.

Figure 6 (top) compares the accuracy (fraction of patterns correctly classified in a batch) of the proposed method for fault classification in an evolving environment with that obtained by performing COMPOSE. Notice that the two methods provide similar performances until time 0.7 (batch 70), and that the proposed method becomes more accurate than COMPOSE after batch 70; furthermore, the 90% confidence intervals identified by the proposed method tend to be smaller than those of COMPOSE.

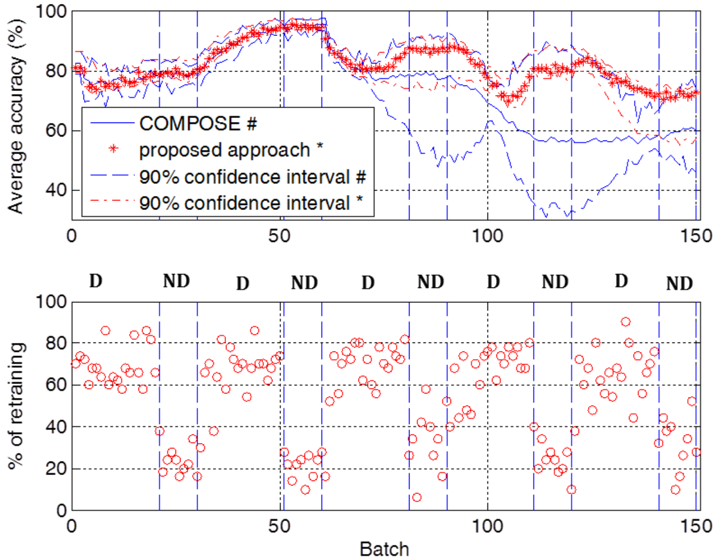

Figure 6 (TOP) Performance of the proposed method and of COMPOSE; Bottom: Percentage of times of classifier retraining (D: batches with concept drift, ND: batches without concept drift)

Figure 6 (bottom) shows the percentage of times in which a concept drift has been detected and, thus, the classifier retrained, over the 50 runs of the experiment. As expected, the drift has been detected more frequently in correspondence of the batches characterized by a concept drift (D) than in correspondence of non-drifted batches (ND). Notice that the presence of the concept drift detector allows remarkably reducing the number of model retrainings with respect to the original COMPOSE, since it does not require to update the classification model each time a new batch of data becomes available.

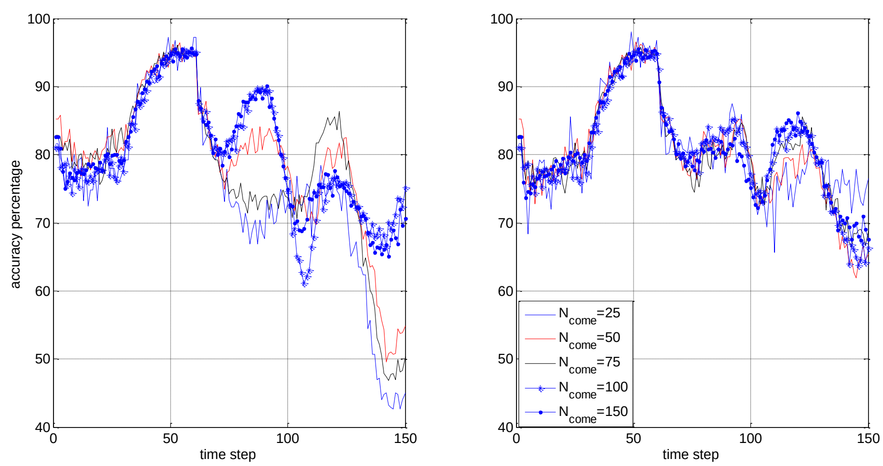

Figure 7 Accuracy of original COMPOSE method and proposed approach considering different numbers of patterns per batch. Left: original COMPOSE; right: proposed approach.

The sensitivity of the two methods with respect to the number of patterns in each batch has been investigated by performing tests with different N_{batch} values. Figure 7 shows that the accuracy of the proposed method is not significantly influenced by N_{batch}, whereas the performance of the original COMPOSE algorithm, tends to become unsatisfactory when the batches are formed by less than 100 patterns. This is due to: *i*) the unnecessary retraining of the classifier *ii*) the difficulty of properly setting COMPOSE parameters α and *CP*, which causes an over shrinkage of the training set.

6.2. The 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 cause 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 [39,40]. 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 dataset, which contains vibrational signals measured by three accelerometers in 72 laboratory tests performed on ball bearings located in two different positions of the powertrain (fan and drive end) [23,40]. The laboratory tests have been performed considering 3 different bearing degradation modes (inner race, outer race and ball defects), 3 different degradation levels ($7, 14, 21$ mil inches defects) and the powertrain working at 4 different loads (0, 1, 2, 3 horsepowers).

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 [41]. Thus, a total number of 720*87 dimensional dataset 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 [42], we have selected three features from the available 87 as input to the diagnostic system. The selection has been performed by applying the feature selection wrapper approach described in [41]. 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 19 batches of data extracted from the Case Western Reserve University bearing dataset become available. In both experiments, we assume that the first available batch of data is formed by all the 180 patterns collected when the powertrain is operating at a load of 0 horsepowers. This batch, which is the only set of patterns for which we assume to have available the labels, i.e. the information on the true class, is used to train the diagnostic model. The remaining 18 batches are formed by 30 patterns collected with the motor working at loads 1, 2 and 3 horsepowers. A batch is always characterized by all patterns collected at a given load.

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. Both 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 fault classifier: the KNN [3] and the SVM [43,44]. Given the more satisfactory performance of SVM on the load 0 horsepowers data, the diagnostic model is built using SVM as classification algorithm.

6.2.1. Results

Figure 8 shows the average performance over the 100 test repetitions of the proposed method and of COMPOSE in the two experiments.

The proposed method is more accurate than COMPOSE, especially as time passes and the number of model retraining increases. Furthermore, thanks to the concept drift detector, it requires less retraining of the classifier (Figure 8, right) with respect to COMPOSE, which requires a new retraining each time a new batch becomes available).

The two tests have also been performed varying the number of patterns per batch. Table 3 reports the average classification accuracy of the proposed method and of COMPOSE.

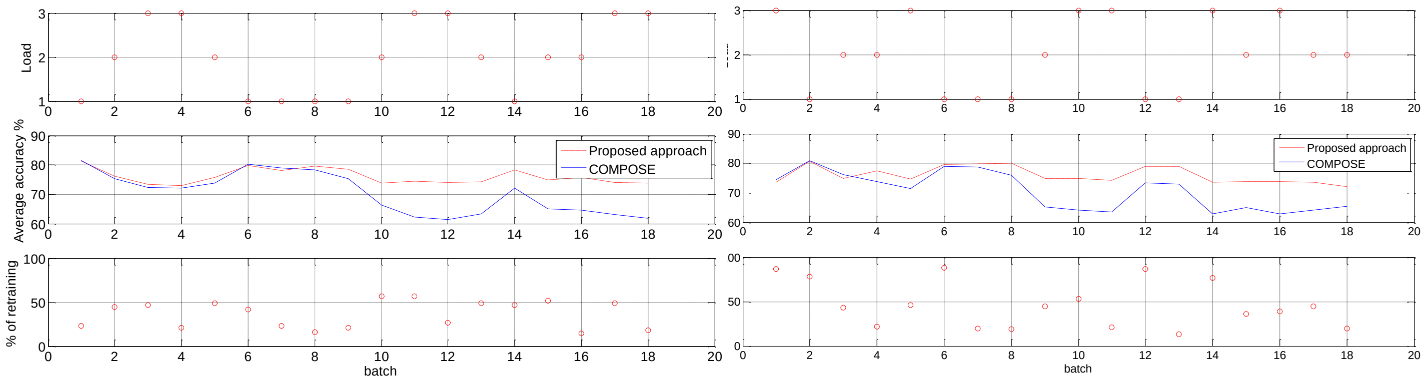

Figure 8 Load sequence (top), average accuracy (middle) and model retraining percentage (bottom) of regular load experiment (left) and random load experiment (right)

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

	Number of pattern in a batch	COMPOSE	PROPOSED METHOD
Regular Load Modifications	10	68.25	76.59
	30	68.48	77.40
	60	78.70	79.19
	90	79.13	80.20
	180	79.52	80.57
Random Load Modifications	10	68.13	76.12
	30	68.25	77.23
	60	77.56	78.16
	90	77.93	78.02
	180	77.77	78.01

Notice that:

1) the performance of both 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 due to the fact that it updates the classifier only when the concept drift is detected in the batch, avoiding unnecessary retraining with small batch of data that reduces the performance.

Finally, it is interesting to observe that in [41] the authors have obtained an average performance of 79.78%, on the same data, but training the classifier using 720 labeled patterns, taken from all the 4 load levels. It seems indeed satisfied that the proposed method, is able to achieve similar performances using only load 0 horsepowers patterns for training.

7. CONCLUSION

This paper 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 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 retrainings and easiness of setting the parameters.

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 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, i.e. 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.

ACKNOWLEDGEMENT

Yang Hu gratefully acknowledges the financial support from China Scholarship Council and Politecnico di Milano (No. 201206110018). The participation of Enrico Zio to this research is partially supported by the China NSFC under grant number 71231001. The participation of Piero Baraldi and Francesco Di Maio is supported by the European Union Project INNovation through Human Factors in risk analysis and management (INNHF, www.innhf.eu) funded by the 7th framework program FP7-PEOPLE-2011-Initial Training Network: Marie-Curie Action.

REFERENCES

[1] Zio E, Gola G. Neuro-fuzzy pattern classification for fault diagnosis in nuclear components. Annals of Nuclear Energy 2006;33:415–26.
[2] Korbicz J. Fault Diagnosis: Models, Artificial Intelligence, Applications. U.S. Government Printing Office; 2004.
[3] Lei Y, Zuo MJ. Gear crack level identification based on weighted K nearest neighbor classification algorithm. Mechanical Systems and Signal Processing 2009;23:1535–47.
[4] Mavromatidis G, Acha S, Shah N. Diagnostic tools of energy performance for supermarkets using Artificial Neural Network algorithms. Energy and Buildings 2013;62:304–14.
[5] Shao M, Zhu X-J, Cao H-F, Shen H-F. An artificial neural network ensemble method for fault diagnosis of proton exchange membrane fuel cell system. Energy 2014;67:268–75.
[6] Selak L, Butala P, Sluga A. Condition monitoring and fault diagnostics for hydropower plants. Computers in Industry 2014;65:924–36.
[7] Di Maio F, Tsui KL, Zio E. Combining Relevance Vector Machines and exponential regression for bearing residual life estimation. Mechanical Systems and Signal Processing 2012;31:405–27.
[8] Unal M, Onat M, Demetgul M, Kucuk H. Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network. Measurement 2014;58:187–96.
[9] Zhao F, Tian Z, Bechhoefer E, Zeng Y. An Integrated Prognostics Method Under Time-Varying Operating Conditions. Reliability, IEEE Transactions on 2015;64:673–86.
[10] Bian L, Gebraeel N. Stochastic methodology for prognostics under continuously varying environmental profiles. Statistical Analysis and Data Mining 2013;6:260–70.
[11] Razavi-Far R, Baraldi P, Zio E. Dynamic Weighting Ensembles for Incremental Learning and Diagnosing New Concept Class Faults in Nuclear Power Systems. Nuclear Science, IEEE Transactions on 2012;59:2520–30.
[12] Kolter JZ, Maloof MA. Dynamic weighted majority: An ensemble method for drifting concepts. The Journal of Machine Learning Research 2007;8:2755–90.
[13] Chen H, Tino P, Rodan A, Yao X. Learning in the Model Space for Cognitive Fault Diagnosis. Neural Networks and Learning Systems, IEEE Transactions on 2014;25:124–36.
[14] Zliobaite I, Bifet A, Pfahringer B, Holmes G. Active Learning With Drifting Streaming Data. Neural Networks and Learning Systems, IEEE Transactions on 2014;25:27–39.
[15] Ben-David S, Blitzer J. Analysis of representations for domain adaptation. Advances in Neural Information Processing Systems 2007;19:137–44.
[16] Daume III H, Marcu D. Domain Adaptation for Statistical Classiers. Journal of Artificial Intelligence Research 2006;26:101–26.
[17] Dyer KB, Capo R, Polikar R. COMPOSE: A Semisupervised Learning Framework for Initially Labeled Nonstationary Streaming Data. IEEE Transactions on Neural Networks and Learning Systems 2014;25:12–26.
[18] Dyer KB, Polikar R. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. Neural Networks (IJCNN), The 2012 International Joint Conference on 2012:1–9.
[19] E IEŽ. Learning under Concept Drift: an Overview. Training 2010;abs/1010.4:1 – 36.
[20] Bose RPJC, van der Aalst WMP, Zliobaite I, Pechenizkiy M. Dealing With Concept Drifts in Process Mining. Neural Networks and Learning Systems, IEEE Transactions on 2014;25:154–71.
[21] Polikar R, Alippi C. Guest Editorial Learning in Nonstationary and Evolving Environments. Neural Networks and Learning Systems, IEEE Transactions on 2014;25:9–11.
[22] Guo B, Menon J, Willette B. Surface Reconstruction Using Alpha Shapes. Computer Graphics Forum 1997;16:177–90.
[23] Case Western Reserve University Bearing Data Center Website. Http://csegroups.case.edu/bearingdatacenter/home 2015.
[24] Samanta B, Al-Balushi KR. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. Mechanical Systems and Signal Processing 2003;17:317–28.
[25] Kuncheva LI, Faithfull WJ. PCA Feature Extraction for Change Detection in Multidimensional Unlabeled Data. IEEE Transactions on Neural Networks and Learning Systems 2014;25:69–80.
[26] Venkatasubramanian V. A review of process fault detection and diagnosis: Part III: Process history based methods. Computers & Chemical … 2003;27:293–311.
[27] Lei Y, Lin J, He Z, Zuo MJ. A review on empirical mode decomposition in fault diagnosis of rotating machinery. Mechanical Systems and Signal Processing 2013;35:108–26.
[28] Lemos A, Caminhas W, Gomide F. Adaptive fault detection and diagnosis using an evolving fuzzy classifier. Information Sciences 2013;220:64–85.
[29] Sakthivel NR, Sugumaran V, Nair BB. Automatic rule learning using roughset for fuzzy classifier in fault categorization of mono-block centrifugal pump. Applied Soft Computing 2012;12:196–203.

[30] Rokach L, Hutter D. Automatic discovery of the root causes for quality drift in high dimensionality manufacturing processes. Journal of Intelligent Manufacturing 2011;23:1915–30.
[31] Horstmann J, Spahr H, Buj C, Münter M, Brinkmann R. Full-field speckle interferometry for non-contact photoacoustic tomography. Physics in Medicine and Biology 2015;60:4045–58.
[32] Aggarwal CC. Data streams: models and algorithms. vol. 31. Springer; 2007.
[33] Gonçalves PM, de Carvalho Santos SGT, Barros RSM, Vieira DCL. A comparative study on concept drift detectors. Expert Systems with Applications 2014;41:8144–56.
[34] Kuncheva LI. Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge and Data Engineering 2013;25:1175–80.
[35] Dries A, Rückert U. Adaptive concept drift detection. Statistical Analysis and Data Mining 2009;2:311–27.
[36] Zweig MH, Campbell G. Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. Clinical Chemistry 1993;39:561–77.
[37] Edelsbrunner H, Mucke EP. 3-dimensional alpha-shapes. ACM Transactions on Graphics 1994;13:43–72.
[38] Kim DS, Seo J, Kim D, Ryu J, Cho CH. Three-dimensional beta shapes. CAD Computer Aided Design 2006;38:1179–91.
[39] Xu Z, Xuan J, Shi T, Wu B, Hu Y. A novel fault diagnosis method of bearing based on improved fuzzy ARTMAP and modified distance discriminant technique. Expert Systems with Applications 2009;36:11801–7.
[40] Wu S-D, Wu P-H, Wu C-W, Ding J-J, Wang C-C. Bearing Fault Diagnosis Based on Multiscale Permutation Entropy and Support Vector Machine. Entropy 2012;14:1343–56.
[41] Piero Baraldi, Francesco Cannarile, Francesco Di Maio EZ. Hierarchical k-nearest neighbours classification and binary differential evolution for fault diagnostics of automotive bearings operating under variable. Reliability Engineering and System Safety 2015:under reviewed.
[42] Emmanouilidis C, Hunter A, MacIntyre J, Cox C. Selecting features in neurofuzzy modelling by multiobjective genetic algorithms. Artificial Neural Networks, 1999 ICANN 99 Ninth International Conference on (Conf Publ No 470) 1999;2:749–54 vol.2.
[43] Langone R, Alzate C, De Ketelaere B, Vlasselaer J, Meert W, Suykens JAK. LS-SVM based spectral clustering and regression for predicting maintenance of industrial machines. Engineering Applications of Artificial Intelligence 2015;37:268–78.
[44] Chang C, Lin C. LIBSVM: a library for support vector machines. Computer 2001:1–30.

Appendix 1: List of features

1. Mean value
2. Kurtosis
3. Skewness value
4. Standard Deviation
5. Crest indicator
6. Clearance indicator
7. Shape indicator
8. Impulse indicator
9. Peak value
10. Minimum Haar Wavelet coefficient
11. Maximum Haar Wavelet coefficient
12. Norm level A3 Daubechies Wavelet transform
13. Norm level D3 Daubechies Wavelet transform
14. Norm level D2 Daubechies Wavelet transform
15. Norm level D1 Daubechies Wavelet Transform
16. Norm Node 1 Symlet6 Wavelet
17. Norm Node 2 Symlet6 Wavelet
18. Norm Node 3 Symlet6 Wavelet

…

28. Norm Node 13 Symlet6 Wavelet
29. Norm Node 14 Symlet6 Wavelet

Appendix 2: Overall procedures of proposed approach

Figure 9 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.

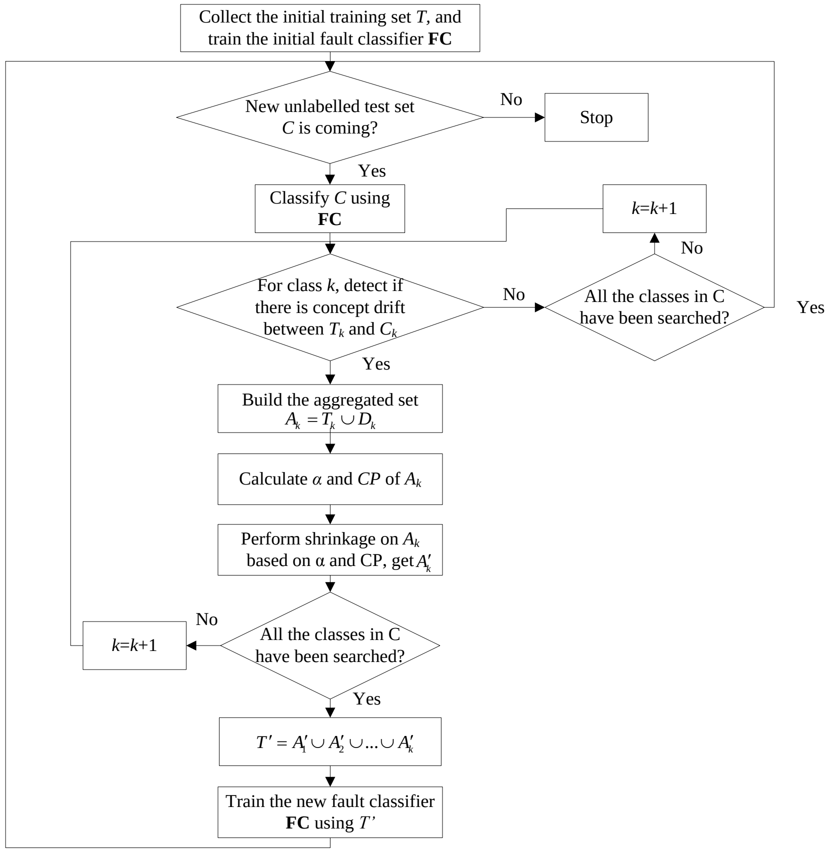

Figure 9 Flow chart of the proposed approach

The details of the procedure are as follows:

1) Collect the initial training set $T = \{\boldsymbol{x}_{\boldsymbol{T}}^{\boldsymbol{i}}, l_{\boldsymbol{T}}^{\boldsymbol{i}}\}, i = 1,2, \ldots, n_{\boldsymbol{T}}, l_T = 1,2, \ldots, n_l$, 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 $C = \{\boldsymbol{x}_{\boldsymbol{C}}^{\boldsymbol{j}}\}, j = 1,2, \ldots, n_{\boldsymbol{C}}$ and use the empirical classifier to label all the patterns in C to output $C = \{\boldsymbol{x}_{\boldsymbol{C}}^{\boldsymbol{j}}, l_{\boldsymbol{C}}^{\boldsymbol{j}}\}$;

3) For class k ($k = 1,2, \ldots, n_l$), perform the α shape based drift detection method (mentioned in subsection 4.2) 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, 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:

   i. Build up the aggregated pattern set $A_k = T_k \cup D_k$, and count the number of patterns in A_k, marked it as $n_{A_{k\_ini}}$;

   ii. Calculate the α value of A_k using equation (4.2) and set the value of *CP* as:

$$CP = 1 - \frac{n_{T_k}}{n_{A_{k\_ini}}}$$

   iii. Perform the shrinkage on A_k using α and *CP*, output the shrunk set A_k';

   iv. Set $D = \emptyset$, 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);

6) Set $k = k + 1$ and go back to step 4) until all the classes are searched;

7) Update the new training set $T' = A_1' \cup A_2' \cup \ldots \cup A_k'$ and retrain the **FC** using T';

8) If new patterns are coming, go back to step 2), otherwise stop the algorithm.