International Journal of Machine Learning and Cybernetics (2023) 14:2925–2944
https://doi.org/10.1007/s13042-023-01810-z

ORIGINAL ARTICLE

CatSight, a direct path to proper multi-variate time series change detection: perceiving a concept drift through common spatial pattern

Arantzazu Flórez[1,2] · Itsaso Rodríguez-Moreno[2] · Arkaitz Artetxe[1] · Igor García Olaizola[1] · Basilio Sierra[2]

Received: 20 October 2022 / Accepted: 21 February 2023 / Published online: 13 March 2023

Abstract

Detecting changes in data streams, with the data flowing continuously, is an important problem which Industry 4.0 has to deal with. In industrial monitoring, the data distribution may vary after a change in the machine's operating point; this situation is known as concept drift, and it is key to detecting this change. One drawback of conventional machine learning algorithms is that they are usually static, trained offline, and require monitoring at the input level. A change in the distribution of data, in the relationship between the input and the output data, would result in the deterioration of the predictive performance of the models due to the lack of an ability to generalize the model to new concepts. Drift detecting methods emerge as a solution to identify the concept drift in the data. This paper proposes a new approach for concept drift detection—a novel approach to deal with sudden or abrupt drift, the most common drift found in industrial processes-, called CatSight. Briefly, this method is composed of two steps: (i) Use of Common Spatial Patterns (a statistical approach to deal with data streaming, closely related to Principal Component Analysis) to maximize the difference between two different distributions of a multivariate temporal data, and (ii) Machine Learning conventional algorithms to detect whether a change in the data flow has been occurred or not. The performance of the CatSight method, has been evaluated on a real use case, training six state of the art Machine Learning (ML) classifiers; obtained results indicate how adequate the new approach is.

Keywords Concept drift · Common spatial pattern · Classification · Data stream

1 Introduction

With the digital transformation in the Industry 4.0 era, the availability of time series data has increased dramatically thanks to process sensing. The rise of real-time connected data presents a great opportunity along with technical challenges, one of which is the modeling of industrial processes while dealing with the constant change of data when the detection of behavioral shifts is crucial. Industrial processes are non-stationary dynamic processes due to various factors such as degradation or operational failures. Those process changes are reflected in the data, but are rarely detected by the model. For model training, the most commonly used methodology consists of historical data-driven offline training, which normally causes a decrease in their performance under changing conditions.

According to recent studies, between 80% and 87% of big data projects fail to generate sustainable solutions [1]. ML algorithms in the real world, and particularly in industry, operate in dynamic environments, with non-stationary data where they need to be able to detect any drift or change in the data distribution and adapt or update the model in order to maintain the performance [2, 3].

The change in the data distribution over time is known as *concept drift* and as defined in [4], a concept drift means that the statistical properties of the target variable change over time in unforeseen ways. The concept drift occurs when the input data changes, which could correspond to a change in the distribution of the data or to a change in the relation between the model input and output. As a consequence, the old data-driven model's performance may decay, and hence it may not be suitable for the new data.

✉ Arantzazu Flórez
aflorez004@ikasle.ehu.eus

Basilio Sierra
b.sierra@ehu.eus

[1] Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Gipuzkoa, Spain

[2] Department of Computer Science and Artificial Intelligence, University of Basque Country, Leioa, Spain

There are different types of concept drift according to [4, 5]:

- Sudden/Abrupt drift: a new concept occurs suddenly
- Gradual drift: a new concept gradually replaces the old one over a period of time
- Incremental drift: an old concept incrementally changes to a new concept
- Recurring concept: an old concept may occur after some time

In this context, in [6] a comprehensive survey that discusses the research constraints and the current state-of-the-art, as well as an updated overview of the different stream mining tasks, such as classification, regression, clustering, and frequent patterns can be found.

As stated by Barros et al. [7], it is quite usual to use a concept drift detection method with a base learner. With the prediction made by the classifier, the drift detector decides whether or not there has been any change in the data distribution. In the last few years, several concept drift detection methods have been proposed in the literature. Based on the classification made by Lu et al. [4], there are three main categories depending on the applied statistical test, namely: Error Rate Based Drift Detection, Data distribution based drift detection and Multiple Hypothesis test Drift Detection. The most well-known concept drift detectors according to the literature [4, 7, 8] are: Drift Detection Method (DDM) [9], Early Drift Detection Method (EDDM) [10], Adaptive Windowing (ADWIN) [11], Statistical Test of Equal Proportions (STEPD) [12], Paired Learners (PL) [13] and EWMA for concept drift detection (ECDD) [14]. In this work, an original approach, consisting of the use of Common Spatial Pattern (CSP) along with ML for discriminating between different concepts, is proposed for sudden drift detection in multivariate time series. CSP is a widely-used method for electroencephalography (EGG) systems to optimally distinguish different classes [15].

Taking this method as a basis—CSP, the proposed approach attempts to differentiate between two concepts in a multivariate signal where a concept drift has occurred. CSP was presented as an extension of the Principal Component Analysis (PCA), but while PCA maximizes the variance of the projected data, CSP maximizes the difference between the variances of two classes. In the case of PCA, the principal component corresponds to the direction of the maximum variability of the data, but it does not assure maximum discrimination between classes, whereas CSP is based on matrix decomposition that maximizes the difference between classes.

The presented approach—CatSight—aims to apply this technique in order to effectively detect sudden drifts in industrial process data. This type of drift is the most common type of concept drift in industrial processes, along with gradual drift [16]. As a matter of fact, the objective of the article is to apply the proposed method to industrial data, thus showing the applicability of CatSight to current problems that may arise in the behavioral changes of machines currently used in industry.

In order to verify how adequate the proposed approach is, a comparison is made with PCA-ML, another feature extractor, and with the conventional ML algorithms. Moreover, to check the suitability of the method, the presented approach has been tested on two publicly available databases, and obtained results compared with state of the art ones; then it has been used on real world industrial data.

Six different base classifiers have been used in the study, and a comparison is made between the *conventional approach* and the new approach presented in this paper—CatSight–. Obtained results confirm the adequateness of the proposed method.

This paper is organized as follows. Section 2 briefly introduces some drift detection methods found in the literature. Section 3.2 introduces the theoretical aspects of the Common Spatial Pattern method. Section 4 details the new approach presented in this work. Section 5 includes the descriptions of the preprocessing steps followed for data preparation, a brief description of ML algorithms used for classification and also includes a brief description of the dataset used. Section 6 presents the results obtained in the different datasets used in the study. In this part, the obtained accuracies have been evaluated. Section 7 presents the results obtained with industrial actual data dataset, which is used as a validator of the proposed approach. Section 8 is a summary of the results obtained. Finally, Sect. 9 draws the conclusions and suggests future work.

2 Related work

Concept drift is a phenomenon that leads to degradation of machine learning performance due to changes in the input data and/or the target variable, in recent years several works can be found in the literature which try to deal with this drawback.

2.1 Machine learning: supervised classification

Zenisek et al. [17] through continuous data flow analysis present an approach based on machine learning to detect drift behavior, and thereby identify the degradation and malfunction of a system. In order to do this, a regression model is used, comparing the estimation made with the model and the real value. Based on the idea that an increased prediction error could indicate a change in concept.

Saurav et al. [18] describe a model based on Recurrent Neural Networks (RNNs) for the detection of anomalies in time series. RNNs are used to make multi-step predictions of the time series, and the prediction errors are used to update the RNN model, as well as to detect anomalies and points of change, thus, a prediction with a large error indicates anomalous behavior or a change of concept in the normal behavior of the time series. Veloso et al. [19] present an extension of Single-pass Self Parameter Tuning SPT, SSPT, a methodology in which the hyperparameters of the model are automatically readjusted when there is a change in concept.

Other authors present classifier ensembles to deal with concept drift detection; in [20] a comparison of several ensemble algorithms is presented, in which 10 different detectors are used; on the other hand, Babüroğlu et al. [21] propose an on-line real-time detector by means of a hybridization of detectors; Wang et al. [22] present a robust novelty detection framework based on ensemble learning.

2.2 Machine learning: unsupervised classification

Liu et al. [23] propose a heuristic method to improve the sensitivity of drift detection, equal intensity k-means space partitioning (EI-kMeans). This approach consists of three components; a greedy equal intensity cluster initialization algorithm, an intensity based cluster amplify-shrink algorithm to unify the cluster intensity ratio and a Pearson's chi-square test-based concept drift detection algorithm. This method is a modification of K-means algorithm drift detector on multi-cluster data; Santos et al. [24] present an empirical method, based on a differential evolution, to tune concept drift detectors in order to improve the obtained accuracy.

Sethi et al. [2] propose the unsupervised Margin Density Drift Detection (MD3) algorithm which tracks the number of samples in the uncertainty region of the classifier, as a metric to detect drift. If a variation is found, the algorithm is retrained.

2.3 Statistics based approaches

In [7] presents a concept drift detection method based on the Wilcoxon rank sum statistical test method. De Lima Cabral et al. [25] also present a drift concept detector based on statistical tests, in this case, the proposed approach is based on Fisher's Exact Test.

Liu et al. [26] propose a drift detector based on Angle Optimized Global Scaling (AOGE) and Principal Component Analysis (PCA). AOGE and PCA analyze the projection angle and variance consecutively, which are subsequently used to identify / detect changes in the objective function.

PCA has been used successfully in studies of dimensionality reduction for multivariate time series analysis with highly correlated variables. For instance, [27] use the PCA to project the high-dimensional data into a principal component projection space before feeding the data to a GAN model. Similarly [28] use PCA to convert multivariate time series datasets into a univariate time series.

The Common Spatial Patterns (CSP) algorithm, first presented in [29] as Fukunaga–Koontz Transform, is a mathematical technique used in signal processing, mainly used in Brain Computer Interface (BCI) applications for electroencephalography (EEG) systems [30–34]. CSP is based on a matrix decomposition method that maximizes the power difference of the two-class signal. This is achieved by maximizing the variance in one class while minimizing the variance of the other class.

Several studies have been conducted by means of CSP. Rodríguez-Moreno et al. [35] present the application of CSP in the shedding light on people action recognition in social robotics. Using this method, a better discrimination between two actions is obtained; this technique allows the signal components that differentiate the actions the most to be extracted. The same authors also present CSP as a feature extraction method that improves the classification task of video activity recognition [36].

In this work, a new approach is presented to deal with temporal data, a new concept drift detection method by means of Common Spatial Pattern. The aim of this algorithm is to filter the data belonging to two populations using the variances to discriminate the signals corresponding to two different targets, finding an optimum spatial filter which reduces the dimensionality of the original signals.

A brief introduction of the Common Spatial Pattern is presented in the following section; a more in-detail presentation can be found in [29].

3 Theoretical aspects

In this paper a new approach to deal with Concept Drift detection is presented. Two main concepts need to be used: Time Series and Common Spatial Patterns.

3.1 Time series

When a variable is measured sequentially in time over or at a fixed interval, known as the sampling interval, it forms a *time series*. The term *univarite time series* refers to a time series that consists of a single observation recorded sequentially in time and *multivariate time series* is used when multiple dependent variables observations are received each time. A time series of length n can be represented by $\{x_t : t = 1, \dots, n\} = \{x_1, x_2, \dots, x_n\}$, which consists of n values sampled at discrete times $1, 2, \dots, n$. When all the observations between specific start and end time are extracted from a time series, the term *time (series) window* is used to refer to it.

The special structure of time series produces unique challenges for machine learning researchers. A consideration due to the special nature of time series is the fact that individual observations are typically highly related with their neighbours in time. Indeed, it is this property that makes most time series excellent candidates for dimensionality reduction.

The main features of many time series are trends and seasonal variations that can be modeled deterministically with mathematical functions in time. A systematic change in a time series that does not appear to be periodic is known as the *trend* and the repeating pattern within any fixed period is called *seasonality*. A *stationary* time series is one whose properties are constant.

It is worth mentioning that Time Windows are the most common way to deal with Time Series analysis and classification. In this paper two very different types of windows are used:

- Short Time Windows (5 or 10 time intervals each) which reflect the temporal evolution are used as individual cases for the Machine Learning classification task.
- A set of the previous time windows is used as a whole dataset; in this case, *n* Time Windows before and *n* after the Concept Drift are selected, and labeled as *before* and *after* for classification purposes. As it will be shown in the Experimental Setup subsection, it is important to notice the difference between *Time Period* and *Time Window*. Both concepts are to be used during the experimental phase, and they can be briefly described as follows:
    - ***T***ime Window: this refers to the consecutive time points that are considered in order to characterize a short time slot; it is used as a summary of the slot itself. Each point could be used individually (1 size Time Window), but it could be computationally expensive, and it may not be that appropriate to show the whole tendency of the series.
    - ***T***ime Period: this is the time elapsed from the beginning to the end of a time series; it refers to all the time steps considered in the performed experiment, and it is composed of several time windows before and the same amount after.

    In this paper, a set of small time windows immediately before and immediately after the concept drift is selected, and used to verify the adequateness of the proposed approach by means of a classification process in which the main characteristics of the small time windows are used as descriptors of each of them. The main novelty of the paper is to use Common Spatial Pattern to obtain a new vision of the data.

3.2 Common spatial pattern description

Common Spatial Pattern method is based on matrix decomposition that maximizes the power difference of the two-class signal. The CSP algorithm requires the information of the class to which the samples belong to calculate the transformation matrix. Thus, CSP tries to find the optimum spatial filters, considering two classes, which maximize the variances of the filtered signals of one of the classes while keeping the variances constant for the other, this way maximizing the difference of the variances between targets. Let $X_{1i}, i = 1, ..., n_1$ and $X_{2i}, i = 1, ..., n_2$ be the signals belonging to two different targets and each element of those lists is a $F \times N$ matrix, having the value of F signals for $t = 1, \dots, N$ time periods. The CSP algorithm calculates a matrix W with optimum spatial filters to transform the original signals X_{ki} (1), where $k = 1, 2$.

$$Z = W^T X_{ki} \tag{1}$$

The first vector of Z contains high variance for the first class signals ($k = 1$) and low variance for the second class signals ($k = 2$), while the last vector contains the opposite, low variance for the first class signals and high variance for the second class signals.

First, to obtain the W matrix, the mean non-centered covariance matrices are calculated (2).

$$R_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} X_{1i} X_{1i}^T; \quad R_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} X_{2i} X_{2i}^T \tag{2}$$

Then, applying the generalized eigen decomposition of the covariance matrices (3), the $W = (\mathbf{w}_1, \dots, \mathbf{w}_F) \in \mathbb{R}^{F \times F}$ projections are calculated, which maximize the function indicated in (4).

$$R_1 \mathbf{w} = \lambda R_2 \mathbf{w} \tag{3}$$

$$\begin{aligned} &\max Tr(W^T R_1 W) \\ &\text{subject to} W^T (R_1 + R_2) W = I \end{aligned} \tag{4}$$

The first and last q vectors are chosen

$W_{CSP} = (\mathbf{w}_1, \dots, \mathbf{w}_q, \mathbf{w}_{F-q+1}, \dots, \mathbf{w}_F)$, where the first q vectors ($j = 1, \dots, q$) obtain large variability for signals that belong to class $k = 1$ ($X_{1i}^T \mathbf{w}_j$) and low variability for signals that belong to class $k = 2$ ($X_{2i}^T \mathbf{w}_j$), and the opposite is obtained with the last q vectors ($j = F - q + 1, \dots, F$). Once the dimensionality of the original signals has been reduced using the W filters, the features are extracted by calculating the variance of each of the output signals Z. Usually the logarithm of the variances is used, hence, the feature vector value for the p-th component of the i-th trial is the logarithm of the normalized variance (5). As mentioned before, the feature vector has $2q$ dimensionality, where q indicates how many vectors of the spatial filter are used in the projection. Exactly, the q first and q last vectors of the aforementioned generalized eigenvectors are used, which yield to the smallest variance for one class and simultaneously, to the largest variance for the other class.

$$f_p^i = log\left(\frac{var(\mathbf{w}_p X_{ki})}{\sum_{p=1}^{2q} var(\mathbf{w}_p X_{ki})}\right) \tag{5}$$

4 Proposed approach

In this paper a new approach is proposed to deal with Concept Drift detection: on the first step, temporal relevant features are projected in a new space by means of an statistical approach called Common Spatial Patterns, and the best among those projected variables are selected and used in the second step to identify the concept drift.

4.1 New approach: CatSight

We present this new approach, called CatSight, as a feature extractor of multivariate time series, and it aims to extract the features that help in the task of differentiating two consecutive time series, having as the final objective the detection of the concept drift in temporal data as is presented in Fig. 1.

The problem is posed as a supervised classification problem in which the label of all the data data is known. As explained in Sect. 3.2, the CSP method is based on matrix decomposition that maximizes the power difference of two class signals. In order to achieve that, first of all, a time period is selected from the multivariate time series $T_t = \{T_{t_0} ... T_{t_n}\}$ where a change in the distribution of the multivariate time series is observed. This change in the distribution is labeled and classified into two different classes, *last concept* and *new concept* and the instant in which this change occurs is called "concept drift". Once the time period is selected, T_{Last} and T_{New}, those two multivariate time series are divided and grouped with different small time windows. Each time window is represented as a NxF matrix, $W_{N \times F}$ where F is the number of features of the multivariate time series and N is the number of samples (time steps) of each time series as shown in Fig. 2.

When the data is prepared, the CSP method is used to transform the data and extract the most relevant features in order to maximize the distance between the two classes. For this, the variance of the transformed data is computed. Then, these features are used as the input to the classifier.

5 Experimental setup

This section presents the different steps performed during the experimental setup.

5.1 Data collection

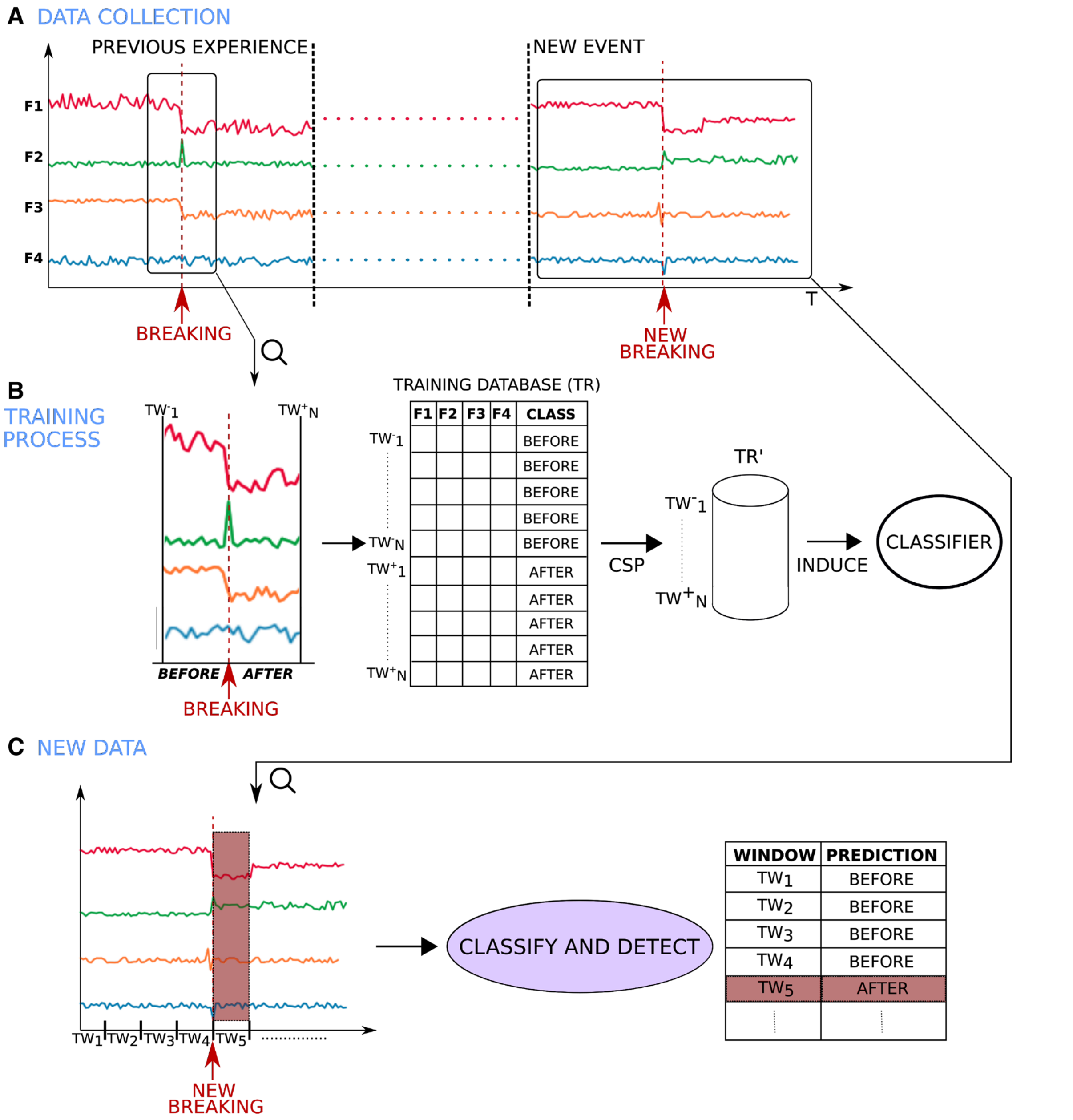
The main objective of this work is to detect the change produced between two states in time series. Therefore, it has been posed as a binary classification problem, in which the objective is to detect if there is any difference between the two states, aiming at differentiating the condition which best fits each section of the evolving data.

This last characteristic is necessary due to the fact that the detection of the concept drift is expected to be between two time intervals named *time period*, i.e., it is aimed at studying whether the characteristics of the signals vary once a concept drift is detected. Consequently, temporal windows are required to compare one state with another; it is worth mentioning that the used windows –composed of only a few time intervals– belong to one class or another, and these are the *cases* of the classification problem.

Considering those requirements, 2 external datasets have been selected from the UCI Machine learning repository and UEA & Kaggle which are summarized in Table 1.

- **EEG Eye State Data Set**: the data was collected by Oliver Roesler [37] in 2013. In this case, the data has been obtained from UCI, a machine learning repository. The dataset belongs to a continuous EEG measurement obtained with the Emotiv Neuroheadset. The subject is recorded with open and closed eyes during 117 s. A total of 14,980 observations were made with 15 attributes (14 electrode measurements and eye states) [38].
- **Water pump sensor data**[1]: data obtained from Kaggle, an online community of data scientists and machine learning practitioners, and composed of water pump data recorded from 52 sensors, which has system failures.
- **Real-world case study**: an industrial use case has been evaluated in order to experimentally validate the proposed method in a real environment. The description is presented in Section. 7

[1] https://www.kaggle.com/nphantawee/pump-sensor-data.



	F1	F2	F3	F4	CLASS
TW^{-}_{1}					BEFORE
					BEFORE
					BEFORE
					BEFORE
TW^{-}_{N}					BEFORE
TW^{+}_{1}					AFTER
					AFTER
					AFTER
					AFTER
TW^{+}_{N}					AFTER

WINDOW	PREDICTION
TW_1	BEFORE
TW_2	BEFORE
TW_3	BEFORE
TW_4	BEFORE
TW_5	AFTER
⋮	⋮

Fig. 1 An overview of the proposed approach. As it could be seen, previous changes are projected using CSP and then used to learn detecting new ones

5.2 Data preprocessing

In this section, some preprocessing steps are adopted to transform and prepare data into a suitable form for the data mining procedure. The preprocessing was performed by applying MinMax normalization and Pearson Correlation filtering (0.8), selecting the most relevant features. Aiming to compare CatSight methodology with its feature extractor counterpart, before splitting the data in different time periods, PCA was applied in the multivarite time series. PCA is a statistical method that converts a set of correlated variables into a set of uncorrelated variables, into a much smaller k principal components.

After that, the pre-processed data is split in different time periods as it is explained in Sect. 4.

5.3 Base learners

Six different classifiers have been selected to perform the experimental phase; they are used to classify the streaming data using both the original sensor data and the projected data obtained with CSP. This work aims to study the

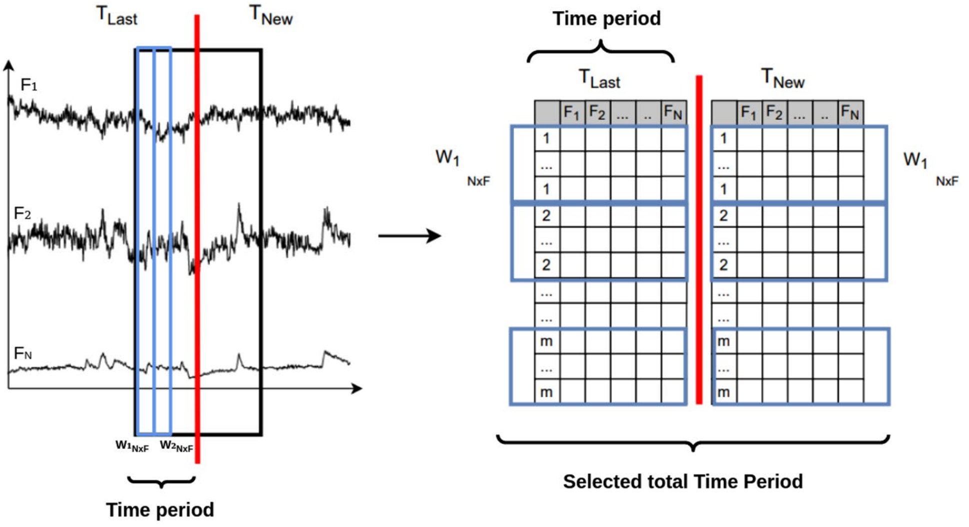


Fig. 2 Multivariate time series with a time period selection, containing data of two different labels. Temporal data is divided in very small time windows which are then used for the learning process

Table 1 List of external datasets and industrial actual data

Data set	Number of instances	Number of attributes	Number of classes
EEG eye state	14,980	15	2
Pump sensor data	1680	52	2
Industrial actual data	7500	23	2

applicability of CSP as a drift detector rather than creating a fine-tuned classifier. For this reason, no fine-tuning methodology has been used and the default values of the scikit-learn[2] (Python module for machine learning) classifiers are used. For each data collection these six classifiers are trained and evaluated by a 5-fold Cross Validation.

- **RandomForest Classifier (RF)**: A RF is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The RF classifier was first defined by Tin Kam Ho in 1995 [39].
- **Support Vector Machine Classifier (SVM)**: This is a supervised learning algorithm developed by Vladimir Vapnik et al. [40]. It constructs a hyperplane in a multidimensional space to separate different classes. The SVM generates an optimal hyperplane iteratively, which is used to minimize an error. The central idea of SVM is to find a maximum marginal hyperplane that best divides the data set into classes.
- **Linear Discriminant Analysis (LDA)**: A classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes'rule. The model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix, [41].
- **KNNeighbors Classifier (KNN)**: K-Nearest Neighbor is a supervised instance-based Machine Learning Algorithm. It ranks values by looking for the most similar (by closeness) data points learned in the training stage and making guesses for new points based on that ranking [42].
- **Classification trees (C4.5)**: The goal of this classifier is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [39].
- **Naive Bayes (NB)**: Performs a classic Bayesian prediction under the assumption that all inputs are independent. Bayesian classifiers use statistical theorems to predict the probabilities of class memberships. It is based on the assumption that the values of the attributes are independent of each other in the calculation of the probabilities of the class known as the class conditional independence [43, 44].

5.4 Comparison

The proposed approach is compared with the conventional method in which the raw data is used after performing the steps described in Sect. 5.2 and with PCA. Data time periods are selected for the training process, explained in Sect. 4, and evaluated by a 5-fold cross validation.

It is worth mentioning that no special computation devices (i.e. GPUs) are needed, and that all the experimental process has been carried out in a conventional PC.

[2] https://scikit-learn.org/stable/.

Table 2 Final datasets

Data set	Case number	Number of attributes	Number of classes	Time period sizes
EEG Eye State	14,980	9	2	360, 180
Pump sensor data	1680	15	2	840, 360, 180
Industrial actual data	7500	9	2	2800, 1800, 1200, 840, 360, 180

6 Experimental results

In this section, the obtained results for the first phase of the experimentation are presented.

6.1 Preprocessing results

Several data preprocessing techniques were performed, such as removing outliers and standardizing data. Correlation filtering was computed to remove the highly correlated features, obtaining the dataset shown in Table 2.

The data has been adapted so that it can be used as input in the classifiers. For this purpose, in each selected dataset, we have obtained the minimum amount of samples that a class has, selecting this minimum value (the minimum value of instances that a class has) as the maximum selected time period. Based on this, and after several experimental tests, we selected a set of window lengths that we considered suitable for the study, which are summarized in Table 2.

As can be observed, the EEG Eye State dataset has 14,980 cases, 9 variables and 2 classes; the selected time period consists of two time periods (T_{Last} and T_{New}), the first selected time period is 360×2, which means that 360 time units are selected before and 360 time units are selected after the Concept Drift occurs. The second selected time period is 180×2 units of time period. Regarding Pump sensor data, there are 1680 cases, 15 variables and 2 classes; used time periods are 840, 360 and 180, hence double time units length each respectively.

With each of the datasets three different time window sizes are used: 5, 10 and 15 time unit sizes have been chosen. Therefore, if a 360 time period is used, for instance, we have 720 time units, which means 144 different time windows (5 time units size) are used for each experiment. Proportionally, 72 windows each of size 10) and 48 (size 15) are used in each experiment.

6.2 Accuracy results and comparison

Tables 3 and 4 present the accuracy scores obtained from the application of the presented approach (CatSight) in selected datasets. In order to assess the effectiveness of the CatSight, a comparison is made with the *conventional way* of classification, which means, in this case, without applying CSP on data, and PCA-ML. In all the studied methods, a 5-fold validation is performed in order to train the base learners.

In the CatSight method, the parameter q indicates the selected features for the classification process, that is; $2 \times q$ features are considered in the training process in CSP. The selected time period corresponds to the number of samples selected $T \in T_{Last}, T_{New}$, and window corresponds to the sub-window.The time window corresponds to the sub-window $w_{N \times F}$ obtained by breaking the multivariate time series into a number of equal size time windows that are used to feed the CSP method.

For each combination of the different parameters of the presented method, it has been evaluated whether the application of CSP may improve the classification results. For this reason, the best results are highlighted in boldface. For every data time window, the best result of the *conventional method* is colored in blue, the best results obtained with PCA-ML are colored in red, whereas, the best results of the new approach are colored in gray. Finally, the green colored values represent the best results obtained from the three methodologies compared in the study.

6.2.1 Water pump sensor data results

In this case, the goal is to analyze whether the new approach is able to identify and improve the classification of the two states of the machine so that it can be established that there has been a change in the operation of the pump.

As can be observed in Fig. 3, the system failed five times during last year, those failures are labeled as *normal*, *broken* and *recovering*. The locations of the breakage points have been identified and those points have been established as reference points for the study so that different time periods were selected from these points. As mentioned before, the minimum amount of data of each class was obtained, selecting this size as the maximum time period.

Each break point is analyzed separately in different stretches, different time periods and windows are established in all the sections as shown in Fig. 4. First of all, the data was processed in order to apply CatSight. Thus, the six classification algorithms were applied in the three compared methodologies and in all the identified streches as a result, the accuracy values of each classifier with a different time period and window was obtained. In the following Table 3,

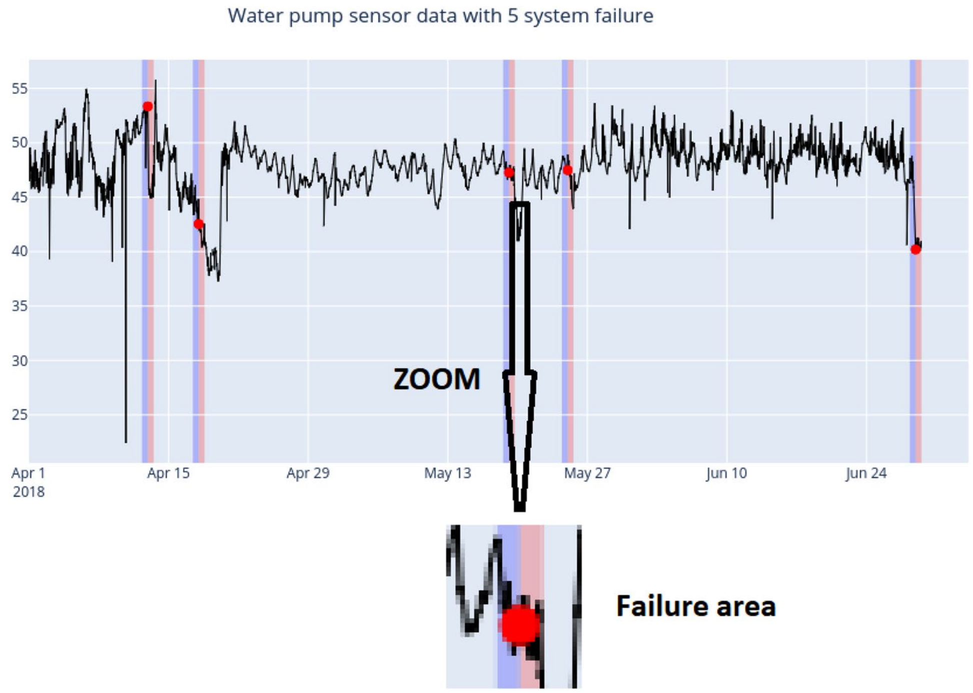


Fig. 3 Water pump sensor data with 5 system failure example. A time period is presented in each failure, delimited between the blue and red lines. The red dot represents the point at which the change in the system has occurred

the mean value of the accuracy is presented, showing the mean value of each classifier in every stretch.

The results in Table 3 show that the best outcome is obtained with CSP, in particular when $q = 2$ or $q = 3$ and with a time period of 840 grouped with a time window of 5, indicating that this configuration is enough to perform the classification. If we compare the results obtained with CatSight and with the conventional way of classification and with PCA-ML, it can be observed that the use of CSP to filter and select temporal variables improves the results of the classification; as a matter of fact, the best result has been obtained using CSP combined with SVM using only 2 (×2) or 3 (×2) variables, with a 0.983 accuracy, using a 840 size time period. It is worth noticing that the result obtained by SVM in the same experiment is 0.834 with the conventional way of classification and 0.833 with PCA-ML. The best result obtained using conventional classifiers without CSP is 0.893, using RF paradigm in a time period of 360 samples and 0.969 with PCA-ML, using SVM with a time period of 360 samples as well.

Regarding the performance of each classifier, it can be seen that in the case of CSP and PCA-ML, the best classifier is SVM, whereas in the conventional case, the best value was obtained by RF and KNN but if we further analyze it, we can see that RF has obtained better results more times than KNN.

In addition, a statistical analysis has been performed to compare all the algorithms against each other. For this purpose, the R package *scmamp*[3] was used which is mainly focused on non-parametric methods and implements Shaffer static [45] and Bergmann and Hommel dynamic corrections [46] for pairwise tests.

As we wanted to compare multiple classifiers, a post-hoc technique was used to visually represent the comparison of the performance of the different algorithms. For that purpose, the critical difference plot was used. The methodology is based on determining whether the performance difference between two algorithms is greater than the critical difference, if this is the case, this is regarded as significantly different [47].

The position of each method in the CD diagram represents their mean ranks across all outcomes of the observations, where the lower ranks indicates that the algorithm performs better more often than its competitors with higher ranks. If two or more algorithms are connected with each other, it means that there is not enough statistical evidence to say that those two algorithms perform differently, whereas it can be said that those that are not connected perform differently.

The results are shown in Fig. 5, where, each algorithm is presented according to its average ranking. On average, SVM.csp and KNN.csp were the best algorithms over all the stretches of the dataset. The horizontal bold line groups the classifiers that show no significant difference, and for

[3] Statistical Comparison of Multiple Algorithms in Multiple Problem; https://github.com/b0rxa/scmamp.

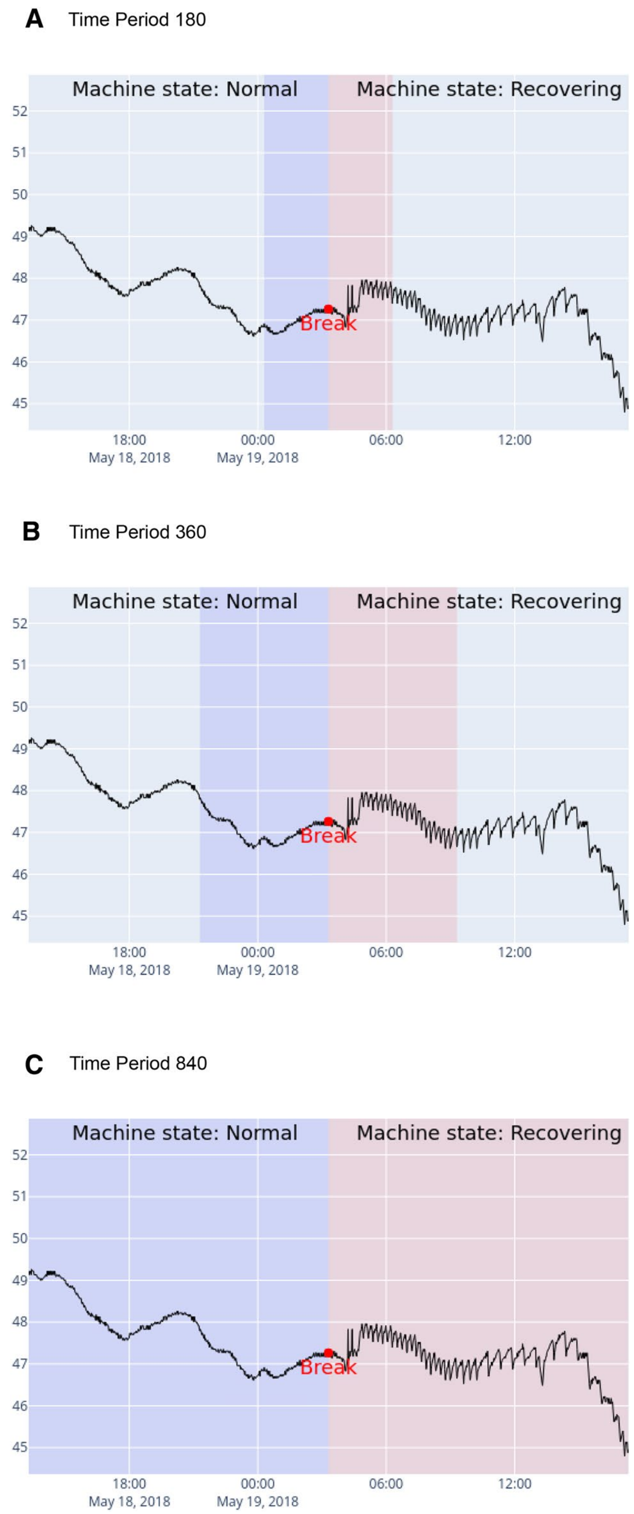


Fig. 4 Water pump sensor stretch example, a failure of the system. Three different time periods are represented, in each of them, the change point is denoted with a red dot

that reason they are grouped together. Moreover, the critical diagram shows that all of the algorithms in which the CSP method was applied before the training process improved their accuracy results over the two methodologies proposed.

6.2.2 Eye state detection results

In the present experiment, we wanted to study if CatSight approach would improve the accuracy results of the conventional method regarding the prediction of the eye open/closed state.

Taking into account the process carried out in the previous experiment, four drift points are established as shown in Fig. 6. As previously done, each stretch is analyzed separately, dividing it into different time periods, as in Fig. 7. The three methodologies were applied in each time period aiming to verify whether the application of the proposed approach improves the accuracy of the classifiers.

The results are summarized in Table 4, where the average of accuracies of the four defined stretches are presented. As Table 4 shows, the application of CSP improves the accuracy rate of the classification models in all the cases studied. The best configuration of the parameters is obtained when the time period is 360 in the case of the CatSight method and with the conventional way of classification, whereas with PCA-ML, the best result is obtained with a time period of 180. If we analyze the value of q, that is, how many new features obtained by CSP transformation are necessary to perform the classification, in this case, the best result is obtained when $q = 3$. The best accuracy is obtained by CatSight using SVM as base classifier (0.955),while the best classifier for conventional classifier is KNN, obtaining a accuracy of (0.744) in both cases. As is shown, the use of CSP improves the obtained results in all the base classifiers.

On the other hand, Fig. 8 displays the critical distance accuracy plots of the multiple algorithms. According to the critical diagram, using CSP in most cases improves the performance of the base classifiers and the methods where PCA is applied, getting the best results with SVM, LDA and KNN. Note that those results also concur with the data presented in Fig. 8.

7 Case study: experimental results with a real industrial dataset

After testing CatSight on publicly available time series data, the proposed methodology was also applied on the data generated from a real-world industrial case study. The studied use case is a centrifugal end suction pump in charge of cooling various components of a metallurgical plant. As shown in Fig. 9, the pump is powered by a coupled induction motor. The motor was a three-phase two-pole induction motor with

Table 3 Pump sensor data CPS accuracy results

Time Period	Window	Method		LDA	RF	KNN	SVM	NB	TREE
840	5	CSP	q=2	0.973	0.974	0.979	**0.983**	0.977	0.968
			q=3	0.974	0.978	0.979	**0.983**	0.977	0.969
			q=5	0.974	0.977	0.980	**0.982**	0.977	0.965
		ML		0.831	**0.878**	0.842	0.834	0.801	0.813
		PCA		0.370	0.772	0.811	**0.833**	0.622	0.772
	10	CSP	q=2	0.960	**0.971**	0.968	0.970	0.964	0.952
			q=3	0.970	0.968	0.973	**0.978**	0.971	0.955
			q=5	0.974	0.973	0.974	**0.977**	0.972	0.955
		ML		0.832	**0.882**	0.854	0.844	0.802	0.851
		PCA		0.389	0.731	0.819	**0.835**	0.627	0.726
	15	CSP	q=2	0.974	0.977	0.979	**0.981**	0.973	0.955
			q=3	0.960	**0.961**	0.957	0.960	0.957	0.943
			q=5	0.965	0.967	0.968	**0.971**	0.966	0.946
		ML		0.830	**0.893**	0.860	0.851	0.803	0.840
		PCA		0.402	0.747	0.814	**0.843**	0.631	0.694
360	5	CSP	q=2	0.951	0.953	0.954	**0.961**	0.951	0.939
			q=3	0.959	0.962	0.964	**0.967**	0.962	0.929
			q=5	0.952	0.955	0.962	**0.968**	0.958	0.922
		ML		0.808	**0.891**	0.805	0.850	0.836	0.853
		PCA		**0.948**	0.930	0.930	0.937	0.933	0.905
	10	CSP	q=2	0.917	0.921	0.940	**0.949**	0.933	0.897
			q=3	0.949	0.952	**0.957**	0.955	0.953	0.912
			q=5	0.943	0.942	**0.952**	0.943	0.951	0.892
		ML		0.806	**0.890**	0.815	0.861	0.836	0.831
		PCA		0.943	0.919	0.946	**0.961**	0.941	0.897
	15	CSP	q=2	0.941	0.926	0.946	**0.948**	0.929	0.880
			q=3	0.934	0.913	0.951	**0.954**	0.938	0.867
			q=5	0.932	0.928	**0.944**	0.936	0.939	0.869
		ML		0.804	**0.888**	0.842	0.883	0.845	0.868
		PCA		0.958	0.930	0.953	**0.969**	0.953	0.905
180	5	CSP	q=2	0.910	0.869	0.910	**0.921**	0.899	0.832
			q=3	0.930	0.916	0.932	**0.933**	0.923	0.878
			q=5	0.924	0.921	**0.934**	0.930	0.925	0.863
		ML		0.742	**0.834**	0.771	0.781	0.795	0.813
		PCA		**0.914**	0.907	0.895	0.876	0.883	0.893
	10	CSP	q=2	0.896	0.888	**0.917**	0.915	0.902	0.875
			q=3	0.889	0.862	**0.922**	0.898	0.920	0.833
			q=5	0.907	**0.913**	0.904	0.903	0.906	0.868
		ML		0.727	**0.842**	0.809	0.814	0.802	0.836
		PCA		0.897	0.896	**0.906**	0.878	0.891	0.873
	15	CSP	q=2	0.812	0.790	0.834	**0.847**	0.806	0.500
			q=3	0.861	0.841	**0.883**	0.873	0.868	0.500
			q=5	0.853	0.865	**0.900**	0.888	0.885	0.500
		ML		0.744	0.814	**0.845**	0.837	0.809	0.817
		PCA		0.906	**0.911**	0.910	0.887	0.901	0.883

a rated power of 75 kW working at 400 V and a rated speed of 3000 r.p.m. at 50 Hz. We would like to point out that this is a practical application of the proposed method to an industrial problem on which the authors are currently working.[4]

The cooling pump is part of a critical system in the plant, as production depends on its correct operation. A machine bearing failure can cause unscheduled production downtime, resulting in economic costs. In the worst case, inadvertent failures can lead to catastrophic damage. For this reason,

[4] https://commons.wikimedia.org/wikiFile:Geothermiekraftwerk_Unterhaching_-_Fernwärmeversorgung.JPG.

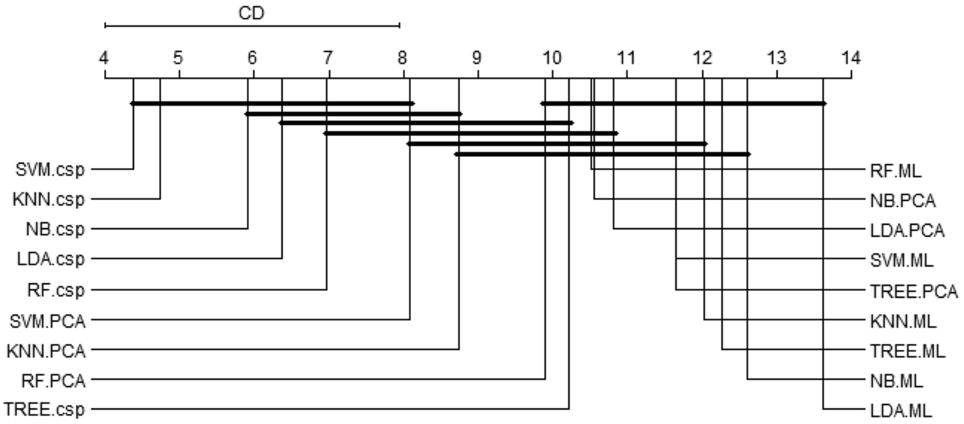


Fig. 5 Statistical comparison of the accuracy of the classifiers using the critical difference diagram on Water Pump data with three methods

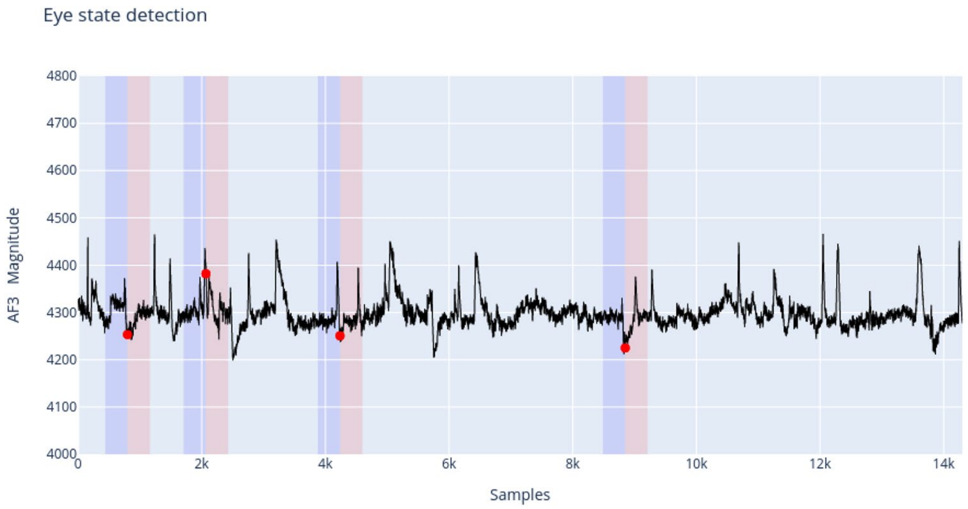


Fig. 6 EEG record example. Every change in the state of the eyes is identified with a red dot and a time period is selected delimited by the red and blue lines

Table 4 Eye state data CSP accuracy results; best results are highlighted in boldface. in each section. the best accuracy results are colored in blue. red and gray. highlighting the best ones in green

Time Period	Window	Method		LDA	RF	KNN	SVM	NB	TREE
360	5	CSP	q=2	0.941	0.931	0.938	**0.945**	**0.945**	0.911
			q=3	0.953	0.948	0.953	**0.955**	**0.955**	0.923
		ML		0.858	0.872	**0.914**	0.894	0.863	0.868
		PCA		**0.695**	0.639	0.644	0.650	0.680	0.642
	10	CSP	q=2	**0.934**	0.931	0.930	0.928	0.923	0.911
			q=3	0.948	0.934	0.954	**0.955**	0.944	0.901
		ML		0.872	0.883	**0.926**	0.907	0.882	0.837
		PCA		**0.719**	0.654	0.687	0.685	0.695	0.67
180	5	CSP	q=2	0.93	0.922	0.932	**0.933**	0.928	0.878
			q=3	0.937	0.935	**0.945**	0.936	0.932	0.874
		ML		0.836	0.859	**0.872**	0.869	0.837	0.817
		PCA		0.719	0.657	0.671	0.665	0.654	0.699
	10	CSP	q=2	0.908	0.887	0.921	**0.926**	0.906	0.834
			q=3	0.932	0.916	0.928	**0.933**	0.931	0.890
		ML		0.85	0.87	0.888	**0.896**	0.871	0.843
		PCA		**0.744**	0.72	**0.744**	0.691	0.666	0.679

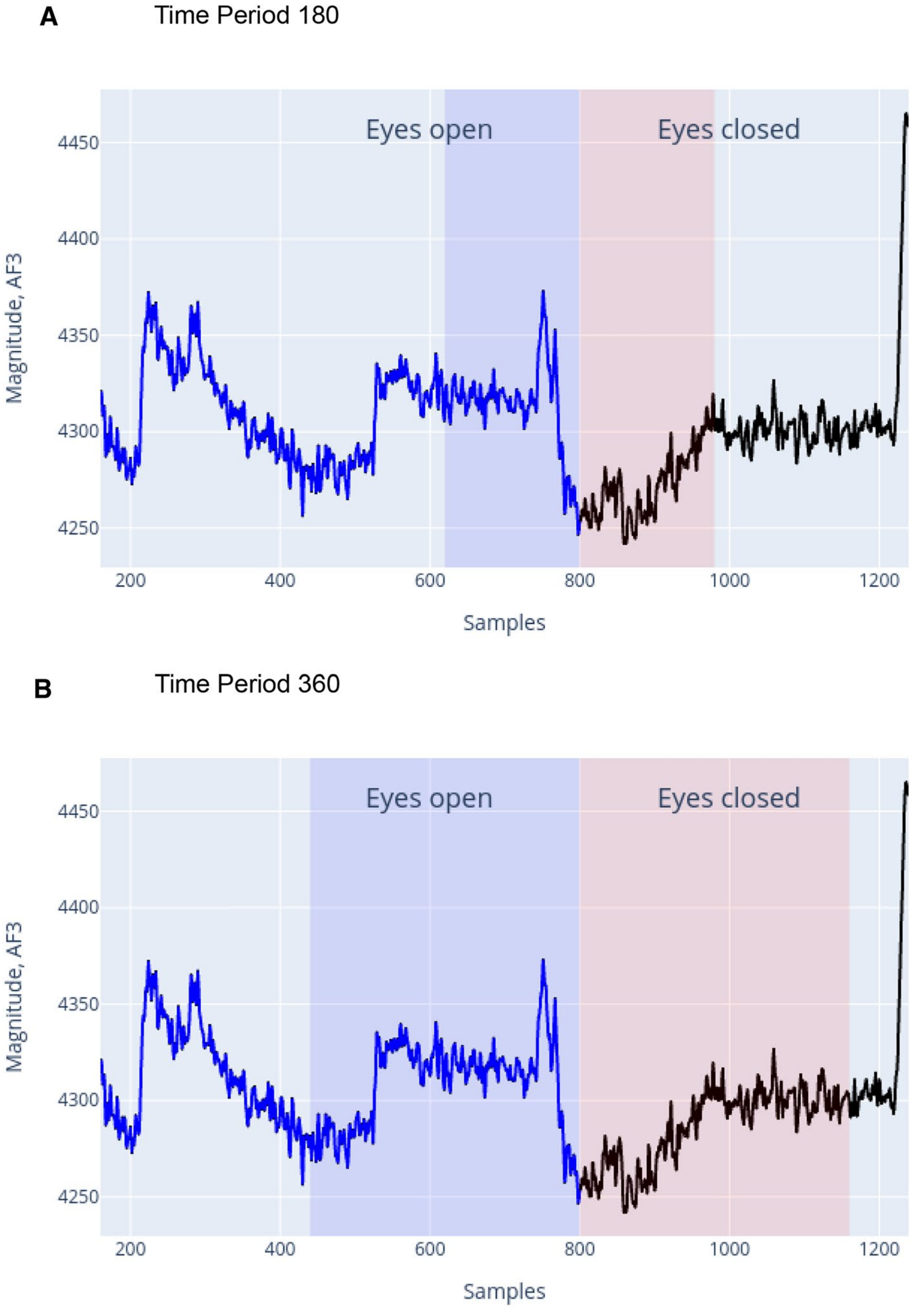


Fig. 7 EEG trace example with different time periods. The selecton of an appropriate time period is an important issue

the system is continuously monitored by several sensors. As shown in Fig. 10, a data logger captures electrical parameters at the inverter input. An accelerometer is also fitted to the motor housing to capture vibration data. Both electrical parameters and vibration data were recorded every six minutes.

The ground truth consists of a bearing ball that gets damaged around a certain time as is shown in Fig. 11, where the red dot corresponds to the moment of the bearing damage. The bearings ensure that the motor shaft is centered, if any of the balls break, the motor shaft may be off-center, which causes a reduction in the remaining useful life (RUL) of the motor. In addition, with an off-center shaft, the motor generates vibrations that propagate towards the load. All this causes the performance of the engine to deteriorate and the sooner the behavioral change is detected, the sooner the necessary actions can be taken. For this reason, we attempted to demonstrate that the methodology presented can be applied in real scenarios and enables to detect the behavioral change of the data.

Taking this *break* point as a reference, the data is divided into two states, namely *normal* and *damaged*, (see Fig. 11), and as in the previous experiments, the minimum amount of a class is obtained in order to establish the stretch size. Time

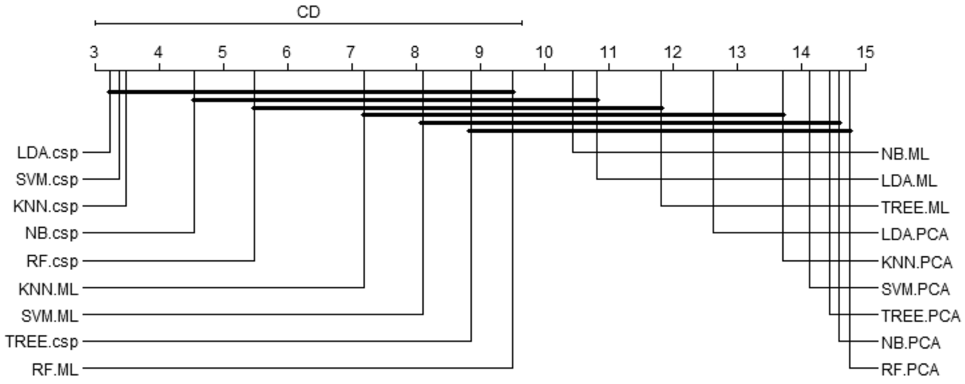


Fig. 8 Accuracy statistical comparison of the methods using the critical difference diagram on Eye state detection data, with three methods

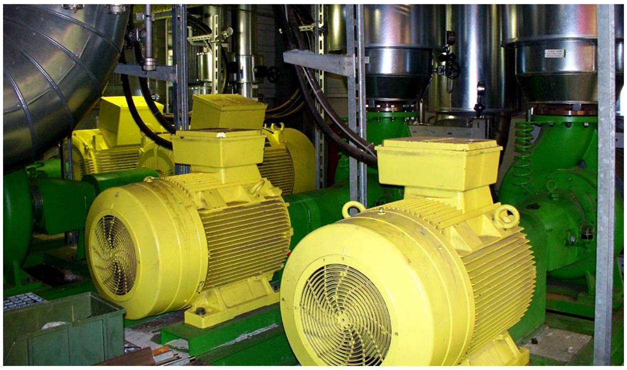


Fig. 9 Long coupled centrifugal pumps[4]

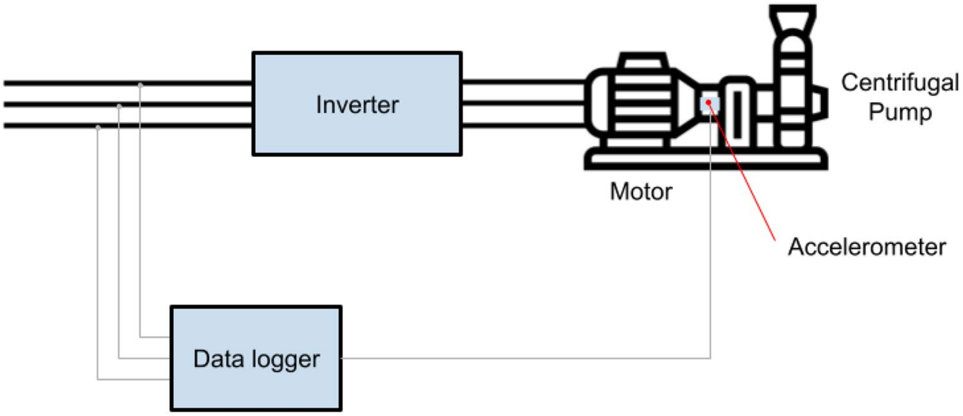


Fig. 10 Setup diagram of the case study

periods were selected taking into account the minimum data amount of one class as was explained before and depicted in Fig. 12. We created temporal sequences of time to feed the three methodologies and taking the presented six classifiers, the three methods are computed in order to compare them and to verify whether CatSight improves the accuracy results.

Analyzing the results in Table 5 and 6, it can be seen that comparing the three methodologies; the CSP based method improves, in general, the accuracies obtained from different algorithms. The best result is obtained by the CSP+TREE algorithm (0.996) using a 2800 time period. The best conventional classifier result is 0.958, obtained by NB; nevertheless, the results of the base classifiers are always improved by means of the CSP approach.

In order to assess the performance of the multiple algorithms that use the analyzed methodologies, we present a post-hoc analysis, in which we use the critical difference diagram. In this case, Fig. 13 illustrates that, for the classification task, all CSP based methods obtained better accuracies. Moreover, analyzing the obtained rank, the best ranked method is SVM-CSP, which is consistent with the observations made from the accuracy tables 5 and 6.

8 Comparison between two external data and industrial actual data

This section summarizes the best values obtained in the three datasets with three methodologies; CatSight, conventional way of classification and PCA-ML. The best results are summarized in Table 7. In general it can be concluded that the new approach outperforms the results comparing

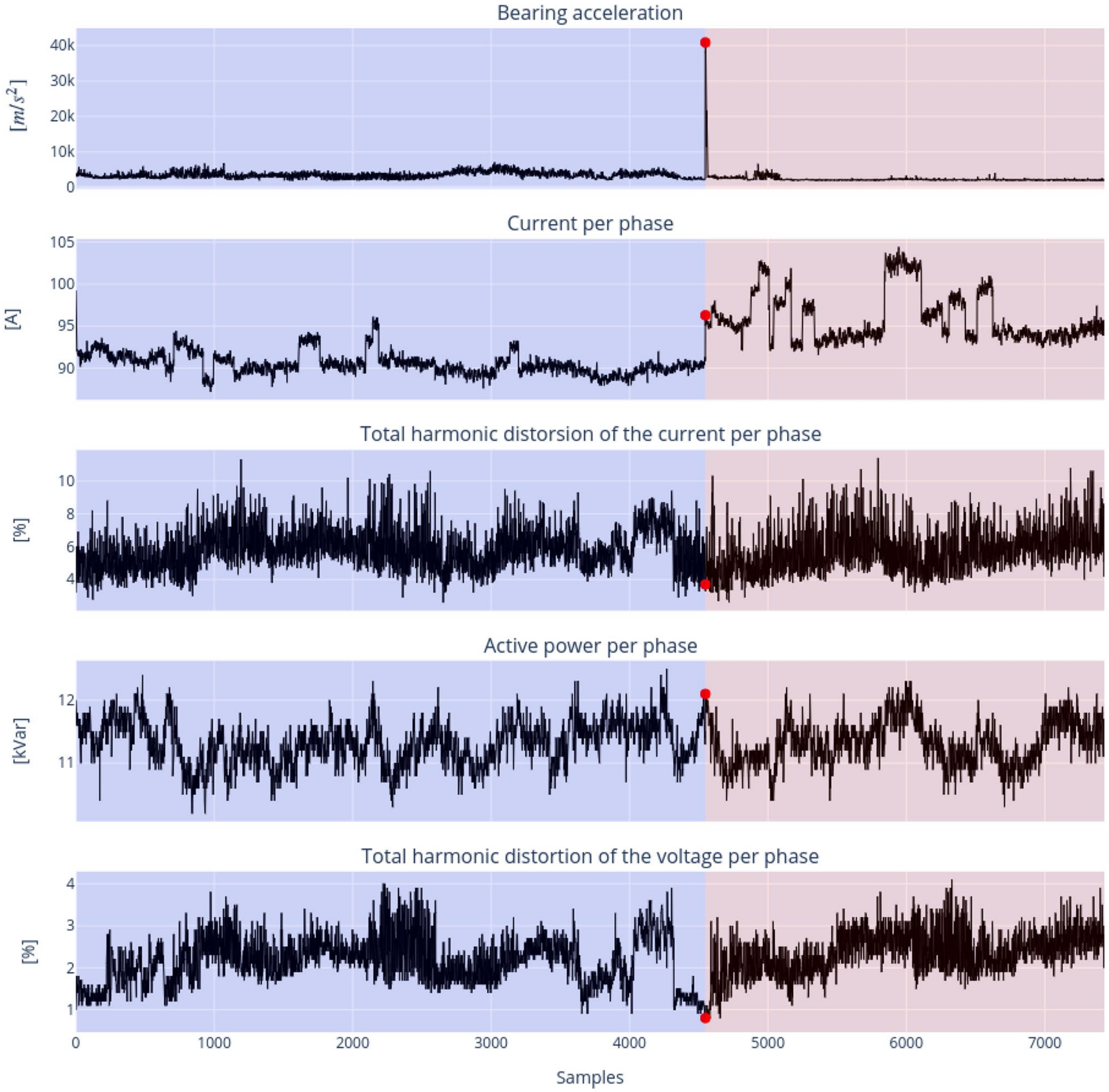


Fig. 11 Industrial actual data feature examples, with the two states delimited by red and blue colors

with the previously proposed methodologies. CSP projects the multivariate temporal data into a clearer space, making the separation of two states higher, and thus achieving better accuracy results. The obtained results using CatSight are more robust as well. Regarding the base classifiers, when CSP is applied, the best performance was obtained using SVM in two of the three datasets used, and with NB in the industrial actual data, followed very closely by SVM; with the conventional methods, there is not a clear winner, RD, KNN and NB have achieved the best results respectively; it is worth noticing that CatSight outperforms the conventional classifiers in all cases. If we compare CatSight with the PCA-ML method, we can see that the results are better than the ones obtained with the conventional way of

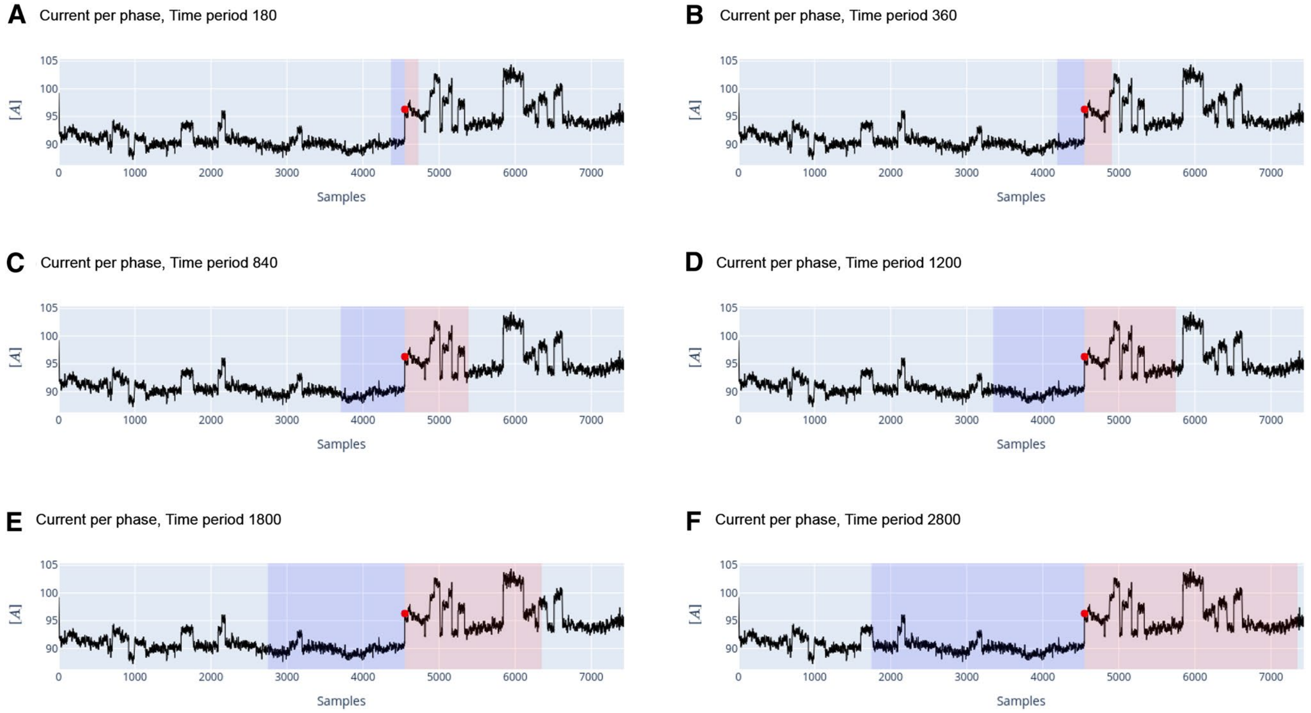


Fig. 12 Current per phase industrial actual data with different time period size

classification, but even so, CSP still outperforms the PCA-ML methods, it performs better and obtains better results.

As it can be seen, the new approach outperforms the standard approaches in mean—of the six Machine Learning classifiers—in the three cases used in this paper: 0.977 for the water pump data, while ML standard approaches obtaines, in mean, 0.848 and using CSP 0.945; on the Eye state detection data, the best mean is again the one achieved by Catsight, 0.948, and the second approach is ML standard classifiers, 0.885, being CSP the wors with 0.707 mean value; in the industrial data the best is Catsight, 0.989, far away from ML standard classifiers (0.850) and CSP (0.841).

9 Conclusion and further work

In this paper, a new approach to deal with concept drift in temporal data is presented, named *CatSight*. It is a combination of two steps; (i) the use of Common Spatial Pattern method to project the multivariate temporal data into a subspace in order to select the most relevant features that separate the two classes in a clearer way and (ii) use of machine learning conventional classification algorithms to detect the change in the data. The drift detector is based on the results of the classifier, using the classification accuracy as a metric.

In order to assess the effectiveness of the method, the CatSight method has been compared with conventional way of classification and the PCA-ML method in three different datasets—two publicly available datasets and one real world industrial dataset-. Experiments show that the CatSight method has a better perfomance among the tested datasets. Generally, higher accuracy rates are obtained with the proposed approach, as an average increase of 10,5% is observed while comparing CatSight with the conventional way of classification. If we analyze the best combination of the CatSight method, it can be stated that the best is the combination of CSP-SVM, which obtains better average accuracy scores than the other methods.

To conclude, in this work, it has been shown that the application of CatSight obtains better discrimination rates between two states in multivariate time series data where a concept drift is observed. Authors believe that this improvement can be applied to several industrial data in industrial problems, as the drift detection capability of the proposed CatSight method has been proven.

Nevertheless, the approach is to be further investigated in order to overcome some of the limitations; to

Table 5 Industrial actual data accuracy results part 1

Time Period	Window	Method		LDA	RF	KNN	SVM	NB	TREE
2800	5	CSP	q=2	0.971	**0.985**	0.983	0.982	0.973	**0.985**
			q=3	0.967	**0.984**	0.976	0.979	0.961	0.982
			q=5	0.966	**0.982**	0.975	0.976	0.966	0.981
		ML		0.628	**0.864**	0.559	0.799	0.673	0.825
		PCA		0.816	0.889	0.857	0.901	0.768	**0.892**
	10	CSP	q=2	0.977	**0.986**	0.982	**0.986**	0.977	0.985
			q=3	0.963	0.981	0.977	0.979	0.967	**0.983**
			q=5	0.96	**0.982**	0.971	0.972	0.965	0.981
		ML		0.677	**0.877**	0.551	0.814	0.657	0.863
		PCA		0.820	0.894	0.871	**0.923**	0.767	0.861
	15	CSP	q=2	0.969	0.984	0.978	0.980	0.97	0.977
			q=3	0.986	0.98	0.985	**0.987**	0.984	0.967
			q=5	0.968	0.979	0.978	0.971	0.965	**0.981**
		ML		0.77	**0.879**	0.56	0.823	0.668	0.841
		PCA		0.821	0.892	0.868	**0.916**	0.768	0.842
1800	5	CSP	q=2	0.986	0.989	0.891	0.989	0.985	**0.990**
			q=3	0.983	0.988	0.984	0.987	**0.989**	0.986
			q=5	0.98	**0.988**	0.982	0.986	0.983	0.989
		ML		0.641	0.878	0.709	0.858	0.817	0.854
		PCA		0.835	0.877	0.841	0.936	0.709	**0.880**
	10	CSP	q=2	0.982	0.988	0.979	0.985	0.985	**0.989**
			q=3	0.988	0.991	0.985	0.989	0.987	**0.996**
			q=5	0.988	**0.991**	0.982	0.986	0.987	**0.991**
		ML		0.678	**0.901**	0.706	0.869	0.813	0.858
		PCA		0.834	0.887	0.845	**0.935**	0.713	0.844
	15	CSP	q=2	0.986	**0.989**	0.988	0.986	0.986	0.989
			q=3	**0.990**	0.989	0.989	0.989	0.983	0.983
			q=5	0.982	0.984	0.983	**0.989**	0.983	0.986
		ML		0.77	**0.922**	0.722	0.877	0.809	0.872
		PCA		0.835	0.877	0.841	**0.936**	0.709	0.861
1200	5	CSP	q=2	0.967	**0.978**	0.971	0.972	0.966	0.969
			q=3	0.971	0.97	0.975	**0.979**	0.973	0.964
			q=5	0.965	**0.982**	0.977	0.980	0.970	0.965
		ML		0.521	0.777	0.805	**0.840**	0.662	0.700
		PCA		0.881	**0.899**	0.809	0.905	0.801	0.870
	10	CSP	q=2	0.974	**0.988**	0.981	0.983	0.976	0.972
			q=3	0.975	0.974	0.975	0.979	**0.980**	0.956
			q=5	0.978	0.978	0.978	**0.980**	0.978	0.967
		ML		0.614	0.811	0.802	**0.859**	0.683	0.785
		PCA		0.895	0.901	0.832	**0.928**	0.806	0.894
	15	CSP	q=2	0.979	**0.985**	0.979	**0.985**	0.975	0.975
			q=3	0.978	**0.988**	0.979	0.986	0.981	0.983
			q=5	0.971	**0.982**	0.978	0.979	0.971	0.975
		ML		0.682	0.855	0.783	**0.859**	0.697	0.770
		PCA		0.901	0.888	0.829	**0.940**	0.815	0.866

continue with this research line, the some further works are envisaged:

On the one hand, the combination of several approaches is to be analysed, similar to the approach proposed in [48], in order to use the appropriate classifier ensemble to each concept drift detection problem. A different approach is also to be tried to help automatically selecting the set of classifiers to be used [49].

On the other hand, streaming data analysis [50] and more real industrial data [51] are also to be investigated.

Table 6 Industrial actual data accuracy results part 2

Time Period	Window	Method		LDA	RF	KNN	SVM	NB	TREE
840	5	CSP	q=2	0.972	0.98	**0.981**	0.976	0.972	0.978
			q=3	0.961	**0.975**	0.966	0.975	0.969	0.969
			q=5	0.971	**0.986**	0.981	0.975	0.969	0.976
		ML		**0.837**	0.787	0.811	0.829	0.82	0.746
		PCA		**0.904**	0.817	0.723	0.874	0.744	0.816
	10	CSP	q=2	0.973	**0.988**	0.982	0.983	0.971	0.985
			q=3	0.972	**0.978**	**0.978**	**0.978**	0.974	0.977
			q=5	0.974	0.977	**0.979**	0.976	0.969	0.977
		ML		**0.842**	0.810	0.785	0.826	0.830	0.811
		PCA		**0.938**	0.832	0.759	0.901	0.772	0.871
	15	CSP	q=2	0.959	0.968	0.968	**0.971**	0.965	**0.971**
			q=3	0.975	0.975	**0.982**	0.978	0.975	**0.982**
			q=5	0.950	0.968	0.976	**0.979**	0.943	0.965
		ML		**0.858**	0.803	0.806	0.824	0.829	0.831
		PCA		**0.947**	0.829	0.747	0.903	0.789	0.833
360	5	CSP	q=2	0.973	0.966	0.973	**0.975**	0.964	0.941
			q=3	0.979	0.98	0.986	**0.989**	0.980	0.950
			q=5	0.959	0.969	0.969	**0.975**	0.972	0.863
		ML		**0.901**	0.880	0.888	0.892	0.871	0.885
		PCA		**0.858**	0.678	0.630	0.772	0.679	0.683
	10	CSP	q=2	0.968	0.977	0.977	**0.986**	0.982	0.945
			q=3	0.961	0.97	0.967	**0.988**	0.973	0.955
			q=5	0.966	0.957	0.977	**0.982**	0.975	0.918
		ML		0.892	0.884	0.894	**0.921**	0.862	0.857
		PCA		**0.898**	0.631	0.593	0.777	0.669	0.660
	15	CSP	q=2	0.950	0.938	0.950	0.950	**0.963**	0.875
			q=3	0.933	**0.967**	0.958	0.950	**0.967**	0.913
			q=5	0.963	0.966	0.972	**0.981**	0.984	0.900
		ML		0.886	0.881	0.902	**0.922**	0.85	0.898
		PCA		**0.928**	0.632	0.635	0.771	0.650	0.709
180	5	CSP	q=2	0.914	**0.918**	0.900	0.909	0.909	0.882
			q=3	0.921	0.930	0.912	**0.948**	0.906	0.909
			q=5	0.959	0.961	0.964	**0.968**	0.957	0.905
		ML		0.658	0.848	0.839	0.823	**0.932**	0.806
		PCA		**0.759**	0.527	0.502	0.536	0.457	0.533
	10	CSP	q=2	0.933	0.925	0.942	**0.958**	**0.958**	0.875
			q=3	0.928	0.933	**0.956**	**0.956**	**0.956**	0.894
			q=5	0.950	**0.958**	0.938	0.954	**0.958**	0.838
		ML		0.648	0.835	0.864	0.841	**0.951**	0.791
		PCA		**0.782**	0.589	0.523	0.554	0.490	0.584
	15	CSP	q=2	**0.925**	0.900	0.913	**0.925**	0.888	0.500
			q=3	0.942	**0.950**	0.883	0.917	0.908	0.500
			q=5	0.963	0.944	0.975	**0.988**	0.969	0.500
		ML		0.638	0.890	0.899	0.872	**0.958**	0.845
		PCA		**0.820**	0.580	0.559	0.556	0.511	0.589

There are also new research lines, such as Change-Point Detection [52] which are to be studied in order to apply them in industrial data.

In this paper, the problem of the drift detector was placed as a balanced classification problem, in which two time windows with the same length were selected. As future work the problem can be analyzed as an unbalanced class data problem, where different time lengths will be selected, so that the change of the concept can be detected as close to the inflection point as possible.

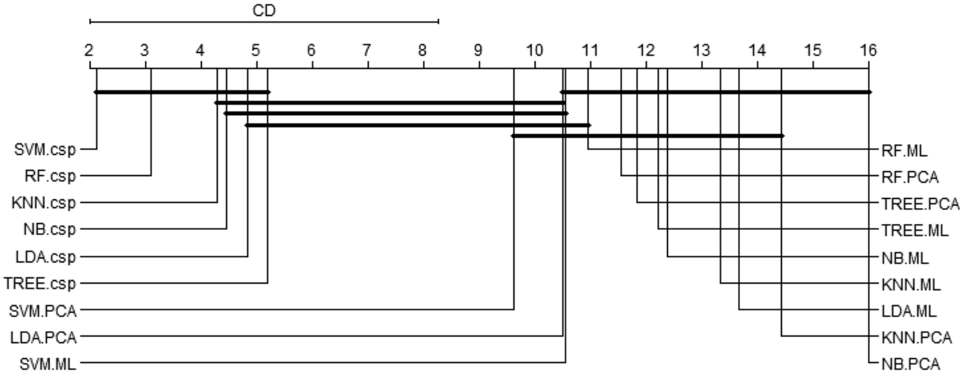


Fig. 13 Accuracy statistical comparison of the methods using the critical difference diagram on industrial actual data, three methods

Table 7 Summary of three dataset best accuracy results

Dataset	Time Period	Window	Method		LDA	RF	KNN	SVM	NB	TREE	*Mean*
Water pump data	840	5	CSP	q=2	0.973	0.974	0.979	**0.983**	0.977	0.968	**0.976**
				q=3	0.974	0.978	0.979	**0.983**	0.977	0.969	**0.977**
		15	ML		0.830	**0.893**	0.869	0.851	0.803	0.840	**0.848**
	360	15	PCA		0.958	0.930	0.953	**0.969**	0.953	0.905	**0.945**
Eye state detection data	360	5	CSP	q=3	0.953	0.948	0.953	**0.955**	**0.955**	0.923	**0.948**
		10		q=3	0.948	0.934	0.954	**0.955**	0.944	0.901	**0.940**
		10	ML		0.872	0.883	**0.926**	0.907	0.882	0.837	**0.885**
	180	10	PCA		**0.744**	0.720	**0.744**	0.691	0.666	0.679	**0.707**
Industial actual data	1800	10	CSP	q=3	0.988	0.991	0.985	0.989	0.987	**0.996**	**0.989**
	180	15	ML		0.638	0.890	0.899	0.872	**0.958**	0.845	**0.850**
	840	15	PCA		**0.947**	0.829	0.747	0.903	0.789	0.833	**0.841**

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work has been partially funded by the Basque Government, Spain, under Grant number IT1427-22; the Spanish Ministry of Science (MCIU), the State Research Agency (AEI), the European Regional Development Fund (FEDER), under Grant numbers TED2021-129488B-I00 and PID2021-122402OB-C21 (MCIU/AEI/FEDER, UE).

References

1. Escobar CA, McGovern ME, Morales-Menendez R (2021) Quality 4.0: a review of big data challenges in manufacturing. J Intell Manuf 2:1–16
2. Sethi TS, Kantardzic M (2017) On the reliable detection of concept drift from streaming unlabeled data. Expert Syst Appl 82:77–99
3. Liu A, Lu J, Zhang G (2020) Diverse instance-weighting ensemble based on region drift disagreement for concept drift adaptation. IEEE Trans Neural Netw Learn Syst 32(1):293–307
4. Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G (2018) Learning under concept drift: a review. IEEE Trans Knowl Data Eng 31(12):2346–2363
5. Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM Comput Surv (CSUR) 46(4):1–37
6. Bahri M, Bifet A, Gama J, Gomes HM, Maniu S (2021) Data stream analysis: Foundations, major tasks and tools. WIREs Data Mining Knowl Discov 11(3):e1405. https://doi.org/10.1002/widm.1405. wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1405
7. de Barros RSM, Hidalgo JIG, de Lima Cabral D.R (2018) Wilcoxon rank sum test drift detector. Neurocomputing 275:1954–1963
8. Gonçalves PM Jr, de Carvalho Santos SG, Barros RS, Vieira DC (2014) A comparative study on concept drift detectors. Exp Syst Appl 41(18):8144–8156
9. Gama J, Medas P, Castillo G, Rodrigues P (2004) Brazilian symposium on artificial intelligence. Springer, Berlin, pp 286–295
10. Baena-Garcıa M, del Campo-Ávila J, Fidalgo R, Bifet A, Gavalda R, Morales-Bueno R (2006) In: Fourth international workshop on knowledge discovery from data streams, vol. 6 pp. 77–86
11. Bifet A, Gavalda R (2007) In: Proceedings of the 2007 SIAM international conference on data mining (SIAM, 2007), pp. 443–448
12. Nishida K, Yamauchi K (2007) In: International conference on discovery science. Springer, Berlin, pp 264–269
13. Bach SH, Maloof MA (2008) in 2008 Eighth IEEE International Conference on Data Mining, pp 23–32

14. Ross GJ, Adams NM, Tasoulis DK, Hand DJ (2012) Exponentially weighted moving average charts for detecting concept drift. Pattern Recogn Lett 33(2):191–198
15. Sadreazami H, Amini M, Ahmad M.O, Swamy M (2021) in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5
16. Sun Z, Tang J, Qiao J, Cui C (2020) in 2020 39th Chinese Control Conference (CCC), pp. 5754–5759
17. Zenisek J, Holzinger F, Affenzeller M (2019) Machine learning based concept drift detection for predictive maintenance. Comput Ind Eng 137:106031
18. Saurav S, Malhotra P, TV V, Gugulothu N, Vig L, Agarwal P, Shroff G (2018) in Proceedings of the acm india joint international conference on data science and management of data , pp. 78–87
19. Veloso B, Gama J, Malheiro B, Vinagre J (2021) Hyperparameter self-tuning for data streams. Inform Fusion 76:75–86
20. de Barros RSM, de Carvalho Santos S.G.T (2019) An overview and comprehensive comparison of ensembles for concept drift. Inform Fusion 52:213–244
21. Babüroğlu ES, Durmuşoğlu A, Dereli T (2021) Novel hybrid pair recommendations based on a large-scale comparative study of concept drift detection. Exp Syst Appl 163:113786
22. Wang B, Wang W, Wang N, Mao Z (2022) A robust novelty detection framework based on ensemble learning. Int J Mach Learn Cybern 2:1–18
23. Liu A, Lu J, Zhang G (2020) Concept drift detection via equal intensity k-means space partitioning. IEEE Trans Cybern 51(6):3198–3211
24. Santos SG, Barros RS, Gonçalves PM Jr (2019) A differential evolution based method for tuning concept drift detectors in data streams. Inf Sci 485:376–393
25. de Lima Cabral DR, de Barros RSM (2018) Concept drift detection based on fisher's exact test. Inform Sci 442:220–234
26. Liu S, Feng L, Wu J, Hou G, Han G (2017) Concept drift detection for data stream learning based on angle optimized global embedding and principal component analysis in sensor networks. Comput Electr Eng 58:327–336
27. Li D, Chen D, Goh J, Ng SK (2018) Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758
28. Zhang Y, Chen Y, Wang J, Pan Z (2021) Unsupervised deep anomaly detection for multi-sensor time-series signals. IEEE Trans Knowl Data Eng 2:2
29. Fukunaga K, Koontz WL (1970) Application of the Karhunen-Loeve expansion to feature selection and ordering. IEEE Trans Comput 4:311–318
30. Ramoser H, Muller-Gerking J, Pfurtscheller G (2000) Optimal spatial filtering of single trial eeg during imagined hand movement. IEEE Trans Rehabil Eng 8(4):441–446
31. Blankertz B, Tomioka R, Lemm S, Kawanabe M, Muller KR (2007) Optimizing spatial filters for robust eeg single-trial analysis. IEEE Signal Process Mag 25(1):41–56
32. Park Y, Chung W (2019) Frequency-optimized local region common spatial pattern approach for motor imagery classification. IEEE Trans Neural Syst Rehabil Eng 27(7):1378–1388
33. Nguyen T, Hettiarachchi I, Khatami A, Gordon-Brown L, Lim CP, Nahavandi S (2018) Classification of multi-class BCI data by common spatial pattern and fuzzy system. IEEE Access 6:27873–27884
34. Xygonakis I, Athanasiou A, Pandria N, Kugiumtzis D, Bamidis P.D (2018) Decoding motor imagery through common spatial pattern filters at the eeg source space. Comput Intell Neurosci **2018**
35. Rodríguez-Moreno I, Martínez-Otzeta JM, Goienetxea I, Rodriguez-Rodriguez I, Sierra B (2020) Shedding light on people action recognition in social robotics by means of common spatial patterns. Sensors 20(8):2436
36. Rodríguez-Moreno I, Martínez-Otzeta J.M, Sierra B, Irigoien I, Rodriguez-Rodriguez I, Goienetxea I (2020) Using common spatial patterns to select relevant pixels for video activity recognition. Appl Sci **10**(22). https://doi.org/10.3390/app10228075. https://www.mdpi.com/2076-3417/10/22/8075
37. Rösler O, Suendermann D (2013)
38. Roesler O (2013) UCI machine learning repository . http://archive.ics.uci.edu/ml
39. Ho TK (1995) in Proceedings of 3rd international conference on document analysis and recognition, vol. 1, IEEE, pp 278–282
40. Vapnik V (1999) The nature of statistical learning theory. Springer Science & Business Media, Berlin
41. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(2):179–188
42. Goldberger J, Hinton GE, Roweis S, Salakhutdinov RR (2004) Neighbourhood components analysis. Adv Neural Inf Process Syst 17:2
43. Zhang H (2004) The optimality of naive bayes. AA 1(2):3
44. Basar MD, Duru AD, Akan A (2020) Emotional state detection based on common spatial patterns of eeg. SIViP 14(3):473–481
45. Shaffer JP (1986) Modified sequentially rejective multiple test procedures. J Am Stat Assoc 81(395):826–831
46. Bergmann B, Hommel G (1988) Multiple hypothesenprüfung/multiple hypotheses testing. Springer, Berlin, pp 100–115
47. Calvo B, Santafé Rodrigo G (2016) scmamp: statistical comparison of multiple algorithms in multiple problems. R J 8:1
48. Ren S, Liao B, Zhu W, Li K (2018) Knowledge-maximized ensemble algorithm for different types of concept drift. Inf Sci 430:261–281
49. Goienetxea I, Mendialdua I, Rodríguez I, Sierra B (2021) Problems selection under dynamic selection of the best base classifier in one versus one: Pseudovo. Int J Mach Learn Cybern 12(6):1721–1735
50. Li C, He C, Zhang H, Yao J, Zhang J, Zhuo L (2022) Streamer temporal action detection in live video by co-attention boundary matching. Int J Mach Learn Cybern 13(10):3071–3088
51. Barrera JM, Reina A, Mate A, Trujillo JC (2022) Fault detection and diagnosis for industrial processes based on clustering and autoencoders: a case of gas turbines. Int J Mach Learn Cybern 2:1–17
52. Hallgren KL, Heard NA, Adams NM (2022) Changepoint detection in non-exchangeable data. Stat Comput 32(6):1–19

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.