Neurocomputing 554 (2023) 126641

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Feature subset selection in data-stream environments using asymmetric hidden Markov models and novelty detection

Carlos Puerto-Santana [a,b,*], Pedro Larrañaga [b], Concha Bielza [b]

[a] Aingura IIoT, San Sebastian/Donotia, Spain
[b] Computational Intelligence Group, Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
Hidden Markov models
Feature subset selection
Data stream
Novelty detection
Feature saliency

ABSTRACT

With the increase of computational power and memory capacity, it is possible to record and analyse lots of features of different nature in real time or in a data stream manner. Nonetheless, in many applications, not all variables may be relevant to explain the nature of the data. Feature subset selection strategies are therefore required to extract the relevant features to understand the data generating process. Much work has been performed in the area of supervised classification problems regarding feature selection, whereas for unsupervised environments, scarce studies have been conducted. In current feature subset selection methodologies for unsupervised data streams, the set of relevant variables are reevaluated or forgotten whenever a new instance or chunk of data arrives: this approach can be computationally expensive or unnecessary. Therefore, in this article, we provide a method to perform feature subset selection in unsupervised data streams. An embedded feature subset selection methodology based on asymmetric hidden Markov models and novel concept discovery strategies is used. This methodology can be used to detect novel concepts in data, and update the set of relevant features when a drift in the data stream is detected. Thus, the relevant variables are updated only when needed. To make this possible, we provide a model for batch and online problems, that is capable of dynamically determining the relevant variables to address feature selection in data streams. Additionally, the model provides domain insights using context-specific Bayesian networks which can be helpful to understand the dynamic process. To validate the proposed methodology, synthetic and real data from ball-bearings are used. Additionally, the proposed methodology is compared with other state of the art methodologies. The proposed method can be used to update the relevant features when needed and is more stable than its competitors.

1. Introduction

Feature subset selection (FSS) is an issue that has been largely reviewed and studied in the literature. The benefits of performing correct feature selection, such as, improving model learning and prediction, simplifying the model, accelerating the computational speed and reducing data storage, are well known [1]. Nevertheless, when working with data streams, selecting features is more difficult as relevant and nonredundant features may change over time [2]. With the development and popularity of paradigms such as the fourth industrial revolution or the Internet of Things, extracting and detecting relevant information in data streams is becoming more relevant. Hence, in this article, an attempt is made to address these problems and provide a solution.

There are three types of FSS methodologies, namely, the filter, wrapper and embedded methodologies. In filtered methodologies, intrinsic information from the data such as correlation, entropy or cross-entropy information is used to discriminate between useful and nonuseful features. This kind of strategy can be fast to implement and execute. Consequently, these strategies are preferred for online analysis. In wrapper methodologies, a model is used to select features; such features are selected to optimize a model score, e.g., accuracy, recall, sensitivity, F1-score or others. Thus, a model must be trained and tested for each feasible set of features, which can be time-consuming and is not suitable for online analysis. In the case of embedded strategies, a feature selection is performed during the model learning phase and feature weights or relevancies are parameters that must be learned. These strategies depend on each model as in the wrapper case. Nonetheless, the learning phase is performed only once; and hence it is faster than the wrapper strategy.

In this article, we will study data streams to observe concept drifts during their development. Three types of concept-drift in the data

* Corresponding author at: Aingura IIoT, San Sebastian/Donotia, Spain.
E-mail address: ce.puerto@alumnos.upm.es (C. Puerto-Santana).

https://doi.org/10.1016/j.neucom.2023.126641
Received 5 May 2022; Received in revised form 15 May 2023; Accepted 24 July 2023
Available online 1 August 2023

stream distribution can occur: real drift, virtual drift and feature drift. A real drift can be understood as a change in the conditional probability distribution of class or clustering variables. A virtual drift is seen as a modification in the joint probability distribution of the variables [3]. A feature drift can be understood as a change in the relevant features that define the class or clustering variable distribution [4]. Any online methodology regarding FSS must address these types of drifts in data.

In this article, we propose an embedded FSS algorithm based on an asymmetric hidden Markov model (HMM) and novel concept discovery. The model uses localized feature saliencies [5] and context-specific Bayesian networks [6] to describe the probability density function (PDF) of relevant features. Our model is an extension of the previous work in [7]. However, features are selected at the hidden state level and the learning phase is performed in an online manner. Additionally, the proposed methodology addresses virtual and feature drift. Real drifts occur after a virtual or feature drift: whenever it arises, a new distribution on the class/cluster variable given those features appears.

Most of the FSS algorithms in unsupervised data streams fail when they try to provide a solution for all the previous properties. Therefore, the main contributions of this work to the state of the art are fivefold:

- A new kind of HMM with localized feature saliencies and context-specific feature interactions.
- An online analysis scheme to update the proposed model when a novel concept appears.
- A methodology to detect feature, virtual and real drifts in training and testing data in an unsupervised manner.
- A model capable of retaining old learned feature subsets and detect them in incoming data instances.
- The proposed methodology is explainable: each detected data trend is explained with its own set of variables and context-specific Bayesian network.

The article is organized as follows: Section 2 reviews the current state of the art regarding FSS for streams, both supervised and unsupervised data streams. Section 3 describes the main theoretical tools used to develop the proposed strategy. Section 4 proposes an offline and online asymmetric HMM with localized feature saliencies. The online version of the model is used to perform an embedded FSS in data streams. Section 5 shows an experimental setup to validate the model capabilities using synthetic and real data streams coming from run-to-failure ball-bearings. Section 6 provides conclusions and further research lines.

2. Related work

In this section, articles related to FSS in streams are reviewed. There are two problems in the literature: the FSS in feature streams and in data streams. The former refers to problems where the number of instances is fixed but the number of features increases over time. Since this problem is out of the scope of this article, its related bibliography is omitted. The latter has a fixed number of features, but the number of instances increases over time. We will focus our attention on FSS in data-streams for unsupervised problems. However, the supervised methodologies are also reviewed for the sake of completeness. Table 1 shows a list of the reviewed articles. There, they are evaluated by five affirmations regarding their capabilities. In this manner, the contributions of this article are stated more clearly. The affirmations are:

A. The method is unsupervised.
B. The method considers feature dependencies.
C. The method has memory regarding previous feature subsets discovered.
D. The method updates the features when necessary and not whenever a new instance or chunk of data arrives.
E. The method searches and acknowledges novel drifts in data.

Table 1
Table of reviewed articles. The articles are compared using five affirmations.

Topic	Name	Reference	A	B	C	D	E
Supervised							
	DXMiner	[8]	–	–	✓	✓	✓
	DXMiner-FI	[9]	–	✓	✓	✓	✓
	DISCUSS	[4]	–	✓	–	–	–
	HEFT-Stream	[10]	–	✓	–	–	✓
	OFS-perceptron	[11]	–	–	–	–	–
	RAC	[12]	–	–	–	–	–
Unsupervised							–
	OLFS-DMM	[13]	✓	–	–	–	–
	Sketching matrix	[14]	✓	✓	–	–	–
	MV-FS	[15]	✓	✓	–	–	–
	FS-K-means	[16]	✓	✓	–	–	–
	DFM-MCFS	[17]	✓	✓	–	–	–
	FSMCP	[18]	✓	–	–	✓	–
	LFS-AsHMM	This article	✓	✓	✓	✓	✓

2.1. *Supervised methodologies*

The earliest work tackling the problem of FSS in data streams can be found in [8]. In such work, the DXMiner was introduced as a model to address infinite data-stream processes with concept drift, feature drift and concept evolution. DXMiner used a finite ensemble of semisupervised K nearest neighbour (K-NN) classifiers that were updated when a new chunk of data was received by the data stream. The FSS algorithm was carried out with a univariate filter FSS algorithm over the new data. The model looks for new classes in unlabelled data using classification boundaries in labelled training data. Later, [9] changed the FSS algorithm and used instead the multicluster FSS model where interactions between features were taken into account for a better FSS procedure (DXMiner-FI). Another filter methodology was found in [4]. The authors proposed dynamic symmetrical uncertainty selection for streams (DISCUSS), where a filter multivariate FSS methodology based on the symmetrical uncertainty score was used to select features. Whenever a new instance was introduced, the score of each feature was updated.

Some authors have attempted to apply boosting or artificial neural networks to learn feature relevancy: Regarding boosting, [10] proposed the heterogeneous ensemble with feature drift for the data stream algorithm (HEFT-Stream). The authors generated an ensemble of classifiers of different types, and each classifier had its own feature space that was generated or updated when a feature drift was detected. FSS was conducted using a multivariate filter methodology, where the features that optimized the symmetrical uncertainty were selected. In the case of neural networks an embedded methodology was found in [11]. The authors proposed an online embedded FSS algorithm based on the perceptron model (OFS-perceptron). A truncation on the weights of the network was set to select the most relevant features. The algorithm learned the weights using an online descent gradient algorithm. In [12] a real-time adaptative component (RAC) based on streaming LASSO regression was proposed for supervised data. In this case, the relevant features were determined with the LASSO regression coefficients. Additionally, an online methodology was proposed to dynamically determine the penalization parameter to select features and forgetting factors.

2.2. *Unsupervised methodologies*

We separate the reviewed articles depending on whether a filter, an embedded or a wrapper methodology was used to select features. We start with embedded methodologies. The authors of [13] proposed an online Dirichlet mixture model with localized feature saliencies (OLFS-DMM). The authors added a latent Bernoulli variable to distinguish between relevant and irrelevant features. The learning process was carried out using an online variational Bayesian method, which updated the model whenever a new instance was introduced. [14] proposed an unsupervised embedded FSS algorithm based on matrix sketching. The algorithm used singular value decomposition on a low rank representation of the full dataset (matrix sketching) and a ranking of components to create a penalized regression problem. Using a regression strategy, the obtained weights were used as a relevancy score to filter the features. In [15], it was assumed that several datasets could be introduced simultaneously or that there are multiple views and the goal was to determine jointly, for each dataset, their relevant features (MV-FS). A nonnegative matrix factorization was carried out for each dataset from each view using a penalized quadratic optimization problem. The output of the optimization was a feature relevancy vector. Finally, [16] proposed an online K-means algorithm with a FSS methodology (FS-K-Means). The authors formulated the K-means optimization problem in terms of indicator matrices, cluster centroid matrices and projection matrices. The projection matrix estimations from the learning phase were used to determine the feature relevancy.

With respect to filter methodologies, [17] proposed a dynamic FSS algorithm that could be used with any model-based clustering (DFM-MCFS). Their idea was to perform a cluster uni variate FSS once a buffer of data was filled. The selected features were used to update a relevancy vector, which indicated the pertinent features. The features that surpassed a given threshold were used in the cluster model. Finally, [18] created an online FSS algorithm based on multicluster structure preservation (FSMCP). First, the membership probabilities of new instances to current clusters were computed. Then, the penalized Kullback–Leibler score was minimized to obtain a set of weights that were used as feature relevancy.

With respect to wrapper methodologies, no articles were found for the case of unsupervised data in data stream environments. A possible reason behind this is that wrapper methodologies require high computational effort and time, which is not feasible in the case of online problems, where the response time of an algorithm must be fast.

As observed form the previous literature review and Table 1, our methodology is novel since it detects and acknowledges novel concept drifts in data, updates the selected feature subsets when it is necessary, it remembers previous learned feature subsets and takes into consideration feature interactions in unsupervised data streams. The only methodology in the state-of-the-art which checks all the previous points is DxMiner-FI [9]. However, such methodology is only applicable for supervised data streams, which implies that they are not comparable. In spite of the previous comments, for the sake of comparison, our proposed methodology is compared in Section 5 with the unsupervised methods of [17] and [18].

3. Theoretical framework

3.1. Hidden Markov models

An HMM is a double chain stochastic model, where one chain is observed, namely $\boldsymbol{X}^{0:T} = (\boldsymbol{X}^0, \dots, \boldsymbol{X}^T)$, with $\boldsymbol{X}^t = (X_1^t, \dots, X_M^t) \in \mathcal{R}^M$ and the other chain is hidden, namely $\boldsymbol{Q}^{0:T} = (Q^0, \dots, Q^T)$. Here, $T+1$ is the length of the data. An HMM can be summarized with the parameter $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) \in \Omega$, where Ω denotes the space of all possible parameters [19], $\mathbf{A} = [a_{ij}]_{i,j=1}^N$ is a matrix representing the transition probabilities between the N hidden states $i, j \in R(Q^t)$ over time, i.e., $a_{ij} = P(Q^{t+1} = j | Q^t = i, \lambda)$; $\mathbf{B}$ is a vector representing the emission probability of the observations given the hidden state, $\mathbf{B} = [b_i(\mathbf{x}^t)]_{i=1}^N$, where $b_i(\boldsymbol{x}^t) = P(\boldsymbol{X}^t = \boldsymbol{x}^t | Q^t = i, \lambda)$ is a probability density function for continuous $\boldsymbol{X}^t$; and $\boldsymbol{\pi}$ is the initial probability distribution of the hidden states, $\boldsymbol{\pi} = [\pi_j]_{j=1}^N$, where $\pi_j = P(Q^0 = j | \lambda)$.

Traditionally, an HMM can solve three problems. First, it can learn the parameter λ, which is usually achieved using the EM algorithm [20]. Second, it can compute the model likelihood of new data, which is solved with the forward–backward algorithm [19]. Third, it can estimate the sequence of hidden states for observed test data, solved with the Viterbi algorithm [21].

3.2. Feature saliency models

Feature saliency models can be seen as embedded feature selection models, since the relevancy of the variables are obtained from the learning process. These models use a set of binary variables, say $\{Z_m\}_{m=1}^M$, which are used as feature relevancy. Each Z_m variable follows a Bernoulli distribution with a parameter ρ_m, which is called the feature saliency of the m-variable. If $\rho_m = 1$, the feature is always relevant. If $\rho_m = 0$, the variable is not relevant. If $\rho_m \in (0,1)$, a threshold $\bar{\rho}$ can be imposed as a decision boundary to determine if a variable is relevant. In the case of HMMs, the Bernoulli variables are added to the emission probabilities, in the general form of:

$$b_i(\boldsymbol{x}^t) = \prod_{m=1}^{M} \left(\rho_m f_{im}(\boldsymbol{x}^t) + (1-\rho_m) g_{im}(\boldsymbol{x}^t) \right) \tag{1}$$

Usually, $f_{im}(\boldsymbol{x}^t)$ represents the relevant distribution or the probability density when the variable X_m is relevant, whereas, $g_{im}(\boldsymbol{x}^t)$ is the probability density when X_m is irrelevant.

There are feature saliency models where Z_m may depend on the hidden variable Q^t; these types of dependencies are represented with an extended set of parameters $\{\rho_{im}\}_{i=1,m=1}^{N,M}$. In these cases, the models are referred to as locally feature saliency models, where the HMM emission probabilities are given as:

$$b_i(\boldsymbol{x}^t) = \prod_{m=1}^{M} \left(\rho_{im} f_{im}(\boldsymbol{x}^t) + (1-\rho_{im}) g_{im}(\boldsymbol{x}^t) \right). \tag{2}$$

Depending on the definitions of $f_{im}(\boldsymbol{x}^t)$ and $g_{im}(\boldsymbol{x}^t)$, the learning algorithm may change from the standard EM algorithm. For example, in [22], the EM algorithm is used to learn the updating formulas of the model; whereas in [5], the authors used stochastic variational methods.

3.3. Hidden state labelling function

In [23] a function $g : R(\boldsymbol{Q}) \to \mathbb{R}$ was introduced to label hidden states of HMMs. The idea behind this function is to assign a real number to each hidden state based on the model parameters. Thus, when a change in a hidden state is observed, not only the change but also its magnitude is registered. As a result, it is no longer necessary to check model parameter values to give a categorical label to a hidden state, but rather to interpret the g function and its output. The g function can be as general as desired; however, it should be computed using the model parameters in an interpretable manner. In this article a new g function is proposed as expressed in Section 4.

3.4. Page sequential test

The Page sequential test is a tool to detect data deviation in time series. Despite its simplicity, it is a strong tool to address and detect anomalous behaviour in data. There are other techniques to detect them (see [24]), such as multidimensional hypothesis tests. Nonetheless, they require labelled data and are highly time-consuming, which may be unfeasible for online applications.

The Page sequential test can be simply stated as follows. Given a time series $s^{0:t} \subset \mathbb{R}$ and a maximum pre-fixed threshold Γ, to detect an anomalous increasing behaviour at time t in the series, the following rule is applied:

$$\begin{cases} \text{If } s^t - \min_{l=0,\dots,t} s^l > \Gamma, & \text{an anomaly is detected} \\ \text{If } s^t - \min_{l=0,\dots,t} s^l \le \Gamma, & \text{no anomaly is detected} \end{cases} \tag{3}$$

For our case study, we use the Bayesian information criterion (BIC) [25] to build the sequence $\{s^t\}_t$. Let $s^t = BIC(\boldsymbol{x}^{0:t})/t$ or the log-likelihood function per unit data. To determine an appropriate threshold Γ for the decision boundary of the test, set c as the maximum allowed probability ratio between $BIC(\boldsymbol{x}^{0:t})/t$ and $\min_{l=0,\dots,t} BIC(\boldsymbol{x}^{0:l})/l$, i.e $c = \frac{BIC(\boldsymbol{x}^{0:t})/t}{\min_{l=0,\dots,t} BIC(\boldsymbol{x}^{0:l})/l}$ and compute its logarithm; in other words: $\Gamma = \log c$. The larger c is, the more tolerant the test is to detect anomalous behaviour.

3.5. Chernoff bounds

The identification of an anomaly can cause false-positive alarms to be triggered. Therefore, it is necessary to determine a reasonable number of anomalies to generate an alarm of undesirable or unknown behaviour. The Chernoff bounds can be used for this purpose [26]. These bounds help us determine the minimum amount of anomalies in a certain window time to declare a change in the data trends. If D_{KL} is the Kullback–Leibler divergence of two Bernoulli distributions, then the Chernoff bounds state that:

$$n^* > \frac{-\ln(1-h)}{D_{KL}(\varphi - e \parallel \varphi)} \tag{4}$$

The previous equation shows that if a sample of n^* instances is measured, with a confidence of $1-h$, if the true proportion of anomalies is φ, the estimated proportion $\hat{\varphi}$ has an error of e with respect to φ. If the estimated proportion $\hat{\varphi}$, surpasses φ, the number of anomalies is important and an alarm related to unknown or undesirable behaviour must be generated.

4. Proposed method

In this section we introduce the proposed model, that is, LFS-AsHMM, which is an asymmetric HMM with local feature saliencies, and explain how to use it in an online manner to perform FSS in data streams. With this model, it is possible to determine the set of relevant features for each hidden state. Then, using context-specific Bayesian networks, the PDF of the relevant features is built. A batch learning algorithm based on the SEM algorithm is presented in Section 4.1, but its online version, to be used in data streams, is presented in Section 4.2. It is relevant to mention that the Viterbi algorithm, in [23] is used to estimate the most likely sequences of hidden states, that can be used to generate the sequences of feature relevancies and state labels.

In the case of the online scenario, an initial LFS-AsHMM is learned and it is updated if a concept drift is detected. A representation of this algorithm is shown in Figure 1. In the *train model* node, the first LFS-AsHMM is trained with the initial observed data using the proposed SEM algorithm. The output of this node is the initial model λ. In the *Novel concept?* node, the current model λ is used to compute the BIC per unit data, perform the Page sequential test and obtain the Chernoff bounds to determine if a concept drift is observed as explained in Section 3. From this node, a Boolean result is obtained that indicates whether it is necessary to update the model. In the *Update model* node, the model is updated (see Section 4.2) if the input Boolean is true, otherwise, the model is not modified. From this node, the model λ is obtained. In the *Viterbi segmentation* node, the Viterbi algorithm is applied to the incoming data to estimate the most likely sequence of hidden states $\boldsymbol{q}^{t:t+L^*}$, where L^* is a parameter for the data windowing scheme discussed in Section 4.2. This sequence goes to the nodes *Compute* $g(i)$ and *Determine FSS*^{t} nodes. The former performs hidden state labelling using the $g(i)$ function explained in Section 4.1.4, as a result the sequence $\{g(q^l)\}_{l=t}^{t+L^*}$ is obtained. The latter computes, the sequence $\{\rho_{q^l m}\}_{l=t}^{t+L^*}$ (see Section 4.1 for the definition of ρ_{im}), which is the feature relevancy over time for variable X_m.

Throughout this section, several symbols and definitions are introduced. Table 2 provides a list of the most relevant symbols and notations for the reader to consult as needed.

4.1. Local feature saliency asymmetric HMM for batch analysis

Here we assume that the emission probabilities are a mixture of Gaussian noise and autoregressive asymmetric linear Gaussian Bayesian networks [23]. Thus, depending on the hidden state, a context-specific Bayesian network and a set of relevant features arise. This model will be known as LFS-AsHMM. If $\boldsymbol{Q}^{0:T}$ or $\boldsymbol{Z}^{0:T}$ is found as a summation

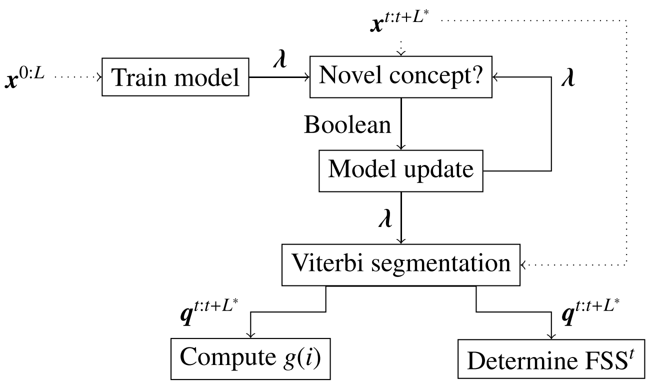


Fig. 1. Pipeline for the online FSS using LFS-AsHMM.

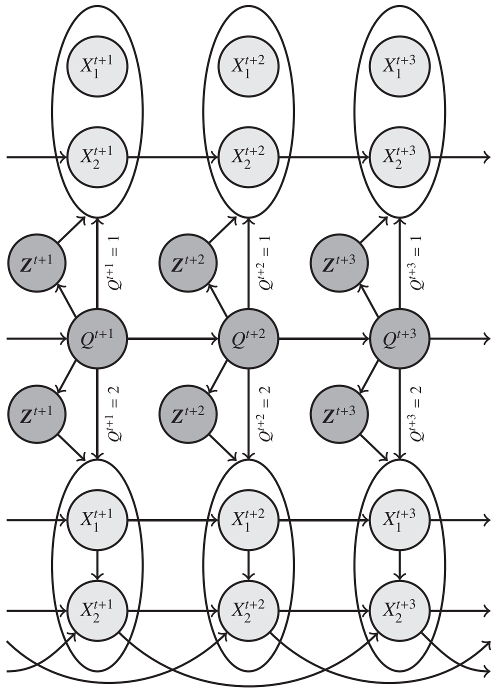


Fig. 2. Example of an LFS-AsHMM as a dynamic Bayesian network.

index, it refers to $\boldsymbol{q}^{0:T} \in R(\boldsymbol{Q}^{0:T})$ or $\boldsymbol{z}^{0:T} \in R(\boldsymbol{Z}^{0:T})$ respectively, where $R(\boldsymbol{F}^{0:T})$ denotes the range of an arbitrary stochastic vector $\boldsymbol{F}^{0:T}$.

In the embedded FSS process, it is assumed that irrelevant features are not affected by changes in hidden states, therefore a Bernoulli vector $\boldsymbol{Z}^t = (Z_1^t, \dots, Z_M^t)$ is introduced in the model with probability:

$$\zeta_i(\boldsymbol{z}^t) := P(\boldsymbol{z}^t | Q^t = i, \lambda) = \prod_{m=1}^{M} \rho_{im}^{z_m^t}(1-\rho_{im})^{(1-z_m^t)}, \tag{5}$$

where $\rho_{im} := P(Z_m^t = 1 | Q^t = i, \lambda)$ for $m = 1, \dots, M$. We assume that the Z_m^t Bernoulli variables have conditional independence between them and that the ρ_{im} parameters change with the hidden state. On the other hand, the irrelevant behaviour is modelled for each variable with a Gaussian distribution that has a mean of ϵ_m and a variance of τ_m^2. The dependency of $\boldsymbol{X}^t$ given $\boldsymbol{Z}^t$ and p^* autoregressive (AR) past values is modelled as:

$$b_i(\boldsymbol{x}^t | \boldsymbol{z}^t) := P(\boldsymbol{x}^t | \boldsymbol{x}^{t-p^*:t-1}, \boldsymbol{z}^t, Q^t = i, \lambda) = \prod_{m=1}^{M} f_{im}(x_m^t)^{z_m^t} g_m(x_m^t)^{(1-z_m^t)}, \tag{6}$$

where $f_{im}(x_m^t) = \mathcal{N}(x_m^t | \boldsymbol{\beta}_{im} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im} \cdot \boldsymbol{d}_{im}^t, \sigma_{im}^2)$ is the PDF for the relevant component, whereas $g_m(x_m^t) = \mathcal{N}(x_m^t | \epsilon_m, \tau_m^2)$ is the PDF of

Table 2
List of relevant symbols and notations.

Symbol	Meaning	Symbol	Meaning
$\boldsymbol{X}^t$	Random vector of observations	p^*	Maximum enabled lag
Q^t	Random variable of unobserved hidden states	$\boldsymbol{\beta}_{im}$	Vector of weights for $\mathbf{pa}^t_{im}$
$\boldsymbol{Z}^t$	Random vector of feature relevancy variables	$\boldsymbol{\eta}_{im}$	Vector of weights for $\mathbf{d}^t_{im}$
$R(\boldsymbol{F}^{0:T})$	Range of a random vector $\boldsymbol{F}^{0:T}$	σ^2_{im}	Variance of variable X_m at $Q^t = i$
$\boldsymbol{A}$	Transition matrix of hidden states	v^t_{im}	Mean of variable X_m at $Q^t = i$
$\boldsymbol{\pi}$	Initial distribution of hidden states	$\mathcal{Q}(\lambda\|\lambda')$	Auxiliary function for the EM algorithm
$f_{im}(x^t_m)$	PDF of X_m at time t, when X_m is relevant	$\mathcal{H}(\lambda\|\lambda')$	Cross entropy of the model λ relative to λ'
$g_m(x^t_m)$	PDF of X_m at time t, when X_m is noise	$LL(\lambda)$	Log-likelihood of the model λ
ϵ_m	Mean for noise PDF for variable X_m	$\zeta^t_i(\boldsymbol{z}^t)$	Probability of random vector $\boldsymbol{z}^t$ at $Q^t = i$
τ^2_m	Variance for noise PDF for variable X_m	$\gamma^t(i)$	A posteriori of $Q^t = i$
ρ_{im}	Feature saliency for variable X_m at $Q^t = i$	$\xi^t(i,j)$	A posteriori of $Q^t = j$ and $Q^{t-1} = i$
p_{im}	AR order for variable X_m at $Q^t = i$	$\psi^t_m(i)$	A posteriori of $Q^t = i$ and $Z^t_m = 1$
$\mathbf{pa}^t_{im}$	Vector of parents of variable X_m at time t	$\phi^t_m(i)$	A posteriori of $Q^t = i$ and $Z^t_m = 0$
$\mathbf{d}^t_{im}$	Vector of AR values of variable X_m at time t	(s)	Iteration of the EM algorithm
$\alpha^t_{p^*}(i)$	Forward variable at $Q^t = i$	$\beta^t_{p^*}(i)$	Backward variable at $Q^t = i$
$g(i)$	Labelling equation of state $Q^t = i$	h	Confidence of the Chernoff bounds
e	Estimation error of the proportion of outliers	φ	Maximum allowed proportion of outliers
n^*	Number of instances for Chernoff bounds	s^t	Page's sequence at time t
Γ	Page's test decision boundary	$\tilde{\rho}$	Decision boundary of feature relevancy
L	Processing data window length	ΔL	New data length
L^*	Maximum window length size	c	Maximum enabled likelihood ratio to set Γ

the noise term. Additionally, $\mathbf{pa}^t_{im} = [1, u^t_{im1}, \dots, u^t_{imk_{im}}]$ and $\boldsymbol{d}^t_{im} = [x^{t-1}_m, \dots, x^{t-p_{im}}_m]$ are vectors with the values of the k_{im} parents of X^t_m in the Bayesian network graph and its $p_{im} \le p^*$ past values. All the previously mentioned parameters can be summarized in the model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, where $\boldsymbol{B}$ is determined by the parameters $(\boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\sigma}^2, \boldsymbol{\epsilon}, \boldsymbol{\tau}^2, \boldsymbol{\rho})$, $\boldsymbol{A}$ is the traditional transition matrix and $\boldsymbol{\pi}$ is the initial distribution.

Fig. 2 shows an example of the new model topology, with two variables/features. When $Q^t = 1$, no probabilistic relationships appear between X^t_1 and X^t_2, and X^t_2 depends on one AR value or (X^{t-1}_2). When $Q^t = 2$, there is a probabilistic dependency of (X^t_2) from (X^t_1), additionally, X^t_1 depends on one AR value. X^{t-1}_1 and X^t_2 depends on two AR values $(X^{t-1}_2$ and $X^{t-2}_2)$. Finally, $\boldsymbol{X}^t$ on both contexts, $Q^t = 1$ and $Q^t = 2$, depends on the binary vector $\boldsymbol{Z}^t$.

From Eqs. (5) and (6) the emission probabilities can be derived:

$$b_i(\boldsymbol{x}^t) := P(\boldsymbol{x}^t|\boldsymbol{x}^{t-p^*:t-1}, Q^t = i, \lambda) = \prod_{m=1}^{M} \rho_{im} f_{im}(x^t_m) + (1-\rho_{im}) g_m(x^t_m), \quad (7)$$

and the full information probability can be written as follows:

$$P(\boldsymbol{q}^{p^*:T}, \boldsymbol{z}^{p^*:T}, \boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \lambda) = \pi_{q^{p^*}} \prod_{t=p^*}^{T-1} a_{q^t q^{t+1}} \prod_{t=p^*}^{T} \zeta_{q^t}(\boldsymbol{z}^t) b_{q^t}(\boldsymbol{x}^t|\boldsymbol{z}^t). \quad (8)$$

4.1.1. E-step

We define the auxiliary function:

$$\mathcal{Q}(\lambda|\lambda') := \sum_{Q^{p^*:T}} \sum_{Z^{p^*:T}} P(\boldsymbol{q}^{p^*:T}, \boldsymbol{z}^{p^*:T}|\boldsymbol{x}^{0:T}, \lambda') \ln P(\boldsymbol{q}^{p^*:T}, \boldsymbol{z}^{p^*:T}, \boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \lambda) \quad (9)$$

From Eq. (9), we can obtain the log-likelihood (LL) of the model λ:

$$\mathcal{Q}(\lambda|\lambda') = \mathcal{H}(\lambda|\lambda') + \ln P(\boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \lambda) = \mathcal{H}(\lambda|\lambda') + LL(\lambda), \quad (10)$$

where

$$\mathcal{H}(\lambda|\lambda') = \sum_{Q^{p^*:T}} \sum_{Z^{p^*:T}} P(\boldsymbol{q}^{p^*:T}, \boldsymbol{z}^{p^*:T}|\boldsymbol{x}^{0:T}, \lambda') \ln P(\boldsymbol{q}^{p^*:T}, \boldsymbol{z}^{p^*:T}|\boldsymbol{x}^{0:T}, \lambda). \quad (11)$$

From Eq. (10), Eq. (11) and [23], it is known that each iteration of the EM algorithm with $\mathcal{Q}(\lambda|\lambda')$ implies improvements in the likelihood function.

By introducing Eq. (8) into Eq. (9), a tractable expression of $\mathcal{Q}(\lambda|\lambda')$ is obtained. It will be useful to find the updating formulas of the model parameters:

$$\begin{aligned}\mathcal{Q}(\lambda|\lambda') = &\sum_{i=1}^{N} \gamma^{p*}(i) \ln(\pi_i^{p^*}) + \sum_{t=p^*}^{T-1}\sum_{i=1}^{N}\sum_{j=1}^{N} \xi^t(i,j) \ln(a_{ij}) \\ &+ \sum_{t=p^*}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} \psi^t_m(i) \ln(\rho_{im} f_{im}(x^t_m)) \\ &+ \sum_{t=p^*}^{T}\sum_{i=1}^{N}\sum_{m=1}^{M} \phi^t_m(i) \ln((1-\rho_{im}) g_m(x^t_m)).\end{aligned} \quad (12)$$

In Eq. (12), we have the latent a posteriori probabilities:

$$\begin{aligned}&\gamma^t(i) := P(Q^t = i|\boldsymbol{x}^{0:T}, \lambda'), && \xi^t(i,j) := P(Q^{t+1} = j, Q^t = i|\boldsymbol{x}^{0:T}, \lambda'),\\ &\psi^t_m(i) := P(Q^t = i, Z^t_m = 1|\boldsymbol{x}^{0:T}, \lambda'), && \phi^t_m(i) := P(Q^t = i, Z^t_m = 0|\boldsymbol{x}^{0:T}, \lambda'),\end{aligned} \quad (13)$$

for $t = p^*, \dots, T$, $i = 1, \dots, N$ and $m = 1, \dots, M$. The E-step consists of estimating these quantities. In the case of $\psi^t_m(i)$, we have:

$$\psi^t_m(i) = P(Q^t = i, Z^t_m = 1|\boldsymbol{x}^{0:T}, \lambda') = \frac{\rho_{im} f_{im}(x^t_m)\gamma^t(i)}{\rho_{im} f_{im}(x^t_m) + (1-\rho_{im}) g_m(x^t_m)}. \quad (14)$$

Note that $\gamma^t(i) = \phi^t_m(i) + \psi^t_m(i)$ for $m = 1, \dots, M$ and $i = 1, \dots, N$. Therefore $\phi^t_m(i) = \gamma^t(i) - \psi^t_m(i)$ and:

$$\phi^t_m(i) = \frac{(1-\rho_{im}) g_m(x^t_m)\gamma^t(i)}{\rho_{im} f_{im}(x^t_m) + (1-\rho_{im}) g_m(x^t_m)}. \quad (15)$$

Now, we state how to estimate $\gamma^t(i)$:

$$\gamma^t(i) = P(Q^t = i|\boldsymbol{x}^{0:T}, \lambda') = \frac{\alpha^t_{p^*}(i)\beta^t_{p^*}(i)}{LL(\lambda')}. \quad (16)$$

In the previous equation the forward variable is $\alpha^t_{p^*}(i) := P(Q^t = i, \boldsymbol{x}^{p^*:t}|\boldsymbol{x}^{0:p^*-1}, \lambda)$ and the backward variable is $\beta^t_{p^*}(i) := P(\boldsymbol{x}^{t+1:T}|Q^t = i, \boldsymbol{x}^{0:t}, \lambda)$. The forward–backward algorithm stated in [23] must be applied to estimate $\alpha^t_{p^*}(i)$ and $\beta^t_{p^*}(i)$. Finally, $\xi^t(i,j)$ can be computed as:

$$\xi^t(i,j) = \frac{\alpha^t_{p^*}(i) a_{ij} b_j(\boldsymbol{x}^{t+1})\beta^{t+1}_{p^*}(j)}{LL(\lambda')}. \quad (17)$$

4.1.2. M-step

The M-step corresponds to optimizing Eq. (12) with respect to the model parameters. The following theorem gives the updating formulas that result from the optimization.

Theorem 1. *Assume there is a current model $\lambda^{(s)}$ from which the E-step has been computed using the formulas in Eq. (13). By maximizing Eq. (12), the resulting parameter $\lambda^{(s+1)}$ can be obtained with the following updating formulas.*

The feature saliencies $\{\rho_{im}^{(s+1)}\}_{m=1}^{M}$ are updated as follows:

$$\rho_{im}^{(s+1)} = \frac{\sum_{t=p^*}^{T} \psi_m^t(i)}{\sum_{t=p^*}^{T} \gamma^t(i)}. \tag{18}$$

The initial distribution $\boldsymbol{\pi}^{(s+1)} = \{\pi_i^{(s+1)}\}_{i=0}^{N}$ is updated as:

$$\pi_i^{(s+1)} = \gamma^{p^*}(i). \tag{19}$$

The transition matrix $\boldsymbol{A}^{(s+1)} = \{a_{ij}^{(s+1)}\}_{i,j=1}^{N}$ is updated as:

$$a_{ij}^{(s+1)} = \frac{\sum_{t=p^*}^{T-1} \xi^t(i,j)}{\sum_{t=p^*}^{T-1} \gamma^t(i)}. \tag{20}$$

The mean and variance, $\{\epsilon_m^{(s+1)}\}_{m=1}^{M}$ and $\{(\tau_m^2)^{(s+1)}\}_{m=1}^{M}$, respectively, from the noise component, are updated as:

$$\begin{aligned} \epsilon_m^{(s+1)} &= \frac{\sum_{t=p^*}^{T} \sum_{i=1}^{N} \phi_m^t(i) x_m^t}{\sum_{t=p^*}^{T} \sum_{i=1}^{N} \phi_m^t(i)} \\ (\tau_m^2)^{(s+1)} &= \frac{\sum_{t=p^*}^{T} \sum_{i=1}^{N} \phi_m^t(i)(x_m^t - \epsilon_m)^2}{\sum_{t=p^*}^{T} \sum_{i=1}^{N} \phi_m^t(i)}. \end{aligned} \tag{21}$$

Denote $v_{im}^t := \boldsymbol{\beta}_{im}^{(s)} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im}^{(s)} \cdot \boldsymbol{d}_{im}^t$ for $m = 1,\ldots,M$, $t = p^*,\ldots,T$ and hidden state $i = 1,\ldots,N$. Then, the relevance parameters $\boldsymbol{\eta}_{im}^{(s+1)} = \{\eta_{imr}^{(s+1)}\}_{r=1}^{p_{im}}$, $\boldsymbol{\beta}_{im}^{(s+1)} = \{\beta_{imk}^{(s+1)}\}_{k=0}^{k_{im}}$ can be updated jointly, solving the following linear equation system:

$$\begin{cases} \sum_{t=p^*}^{T} \psi_m^t(i) x_m^t = \sum_{t=p^*}^{T} \psi_m^t(i) v_{im}^t \\ \sum_{t=p^*}^{T} \psi_m^t(i) x_m^t u_{im1}^t = \sum_{t=p^*}^{T} \psi_m^t(i) u_{im1}^t v_{im}^t \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \sum_{t=p^*}^{T} \psi_m^t(i) x_m^t u_{imk_{im}}^t = \sum_{t=p^*}^{T} \psi_m^t(i) u_{imk_{im}}^t v_{im}^t \\ \sum_{t=p^*}^{T} \psi_m^t(i) x_m^t x_m^{t-1} = \sum_{t=p^*}^{T} \psi_m^t(i) x_m^{t-1} v_{im}^t \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \sum_{t=p^*}^{T} \psi_m^t(i) x_m^t x_m^{t-p_{im}} = \sum_{t=p^*}^{T} \psi_m^t(i) x_m^{t-p_{im}} v_{im}^t \end{cases} \tag{22}$$

The previous system of equations and its solution can be written in a more compact manner: Let $\boldsymbol{\theta}_{im} = [\boldsymbol{\beta}_{im}, \boldsymbol{\eta}_{im}]^\top$ be a column vector, $\boldsymbol{U}_{im} = [\mathbf{pa}_{im}, \boldsymbol{d}_{im}]$ and $\boldsymbol{U}_{im} \circ \boldsymbol{\psi}_{im}$ denote the pointwise column multiplication of every column of $\boldsymbol{U}_{im}$ by $\boldsymbol{\psi}_{im}$. The updating formula of $\boldsymbol{\theta}_{im}$ in Eq. (22) can be written as:

$$\begin{aligned} (\boldsymbol{U}_{\boldsymbol{im}} \circ \boldsymbol{\psi}_{im})^\top \boldsymbol{X}_m &= (\boldsymbol{U}_{\boldsymbol{im}} \circ \boldsymbol{\psi}_{im})^\top \boldsymbol{U}_{\boldsymbol{im}} \boldsymbol{\theta}_{im} \\ \boldsymbol{\theta}_{im}^{(s+1)} &= ((\boldsymbol{U}_{\boldsymbol{im}} \circ \boldsymbol{\psi}_{im})^\top \boldsymbol{U}_{\boldsymbol{im}})^{-1} (\boldsymbol{U}_{\boldsymbol{im}} \circ \boldsymbol{\psi}_{im})^\top \boldsymbol{X}_m \end{aligned} \tag{23}$$

where $\top$ stands for the matrix transpose operation. Set $\hat{v}_{im}^t := \boldsymbol{\beta}_{im}^{(s+1)} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im}^{(s+1)} \cdot \boldsymbol{d}_{im}^t$. Then, $\{(\sigma_{im}^2)^{(s+1)}\}_{i,m=1}^{N,M}$ can be updated as:

$$(\sigma_{im}^2)^{(s+1)} = \frac{\sum_{t=p^*}^{T} \psi_m^t(i)(x_m^t - \hat{v}_{im}^t)^2}{\sum_{t=p^*}^{T} \psi^t(i)}. \tag{24}$$

It is worth noting that, from Eq. (22) for each variable $m = 1,\ldots,M$ and hidden state $i = 1,\ldots,N$, the size of the linear system will depend on the number of parents and AR values; the longer the list of dependencies is, the larger the linear system.

4.1.3. Structural EM (SEM)

In this article we use the greedy-forward algorithm proposed in [23] to search the space of possible graphical models. However, from this model, it is plausible to believe that if a variable is Gaussian noise, it should not be considered in any explanatory graphical model. Therefore, we impose a restriction during the search of structures so that no noise variable is added to any context-specific Bayesian network. The restriction consists of omitting any possible arc coming into or from variables X_m that fulfil the following condition: $\rho_{im} \leq \tilde{\rho}$, where $\tilde{\rho} \in [0,1)$ is a threshold that determines which variables are relevant. Recall that the opposite of this assumption is not true, i.e., if a variable does not have any relationship with any other variable in a context-specific Bayesian network, it does not mean that it is Gaussian noise under our relevance definition.

4.1.4. Hidden state labelling

It is well known that a linear Gaussian Bayesian network can be expressed as a multivariate Gaussian distribution [27]. Let $\boldsymbol{\mu}_i = \{\mu_{im}\}_{m=1}^{M}$ be the mean vector corresponding to the linear Gaussian Bayesian network related to $Q^t = i$. For these models, we set the labelling g equation as follows (Section 3.3):

$$g(i) = \sum_{m=1}^{M} |\mu_{im}| \chi_{\{\rho_{im} \geq \tilde{\rho}\}} + \sum_{m=1}^{M} |\epsilon_m| \chi_{\{\rho_{im} < \tilde{\rho}\}}, \tag{25}$$

where χ_δ is the indicator function. It is one if the condition δ is met and zero otherwise. This g function adds the true information from all the features, the greater the g value is, the greater the magnitude of the mean of the observed data.

4.2. Localized feature saliency asymmetric HMM for online analysis

It will be assumed here that at the beginning of the data stream, there is only one concept and therefore only one hidden state will be learned. The idea is to increase the number of hidden states whenever a concept drift is detected in the data stream. A concept drift is detected when Page's sequential test and the Chernoff bounds give evidence of anomalous data. Due to the flexibility of context-specific feature saliencies, the feature saliencies learned for each hidden state are not lost and the relevancy of each feature can be tracked during the online analysis.

4.2.1. Online learning

Online processing consists of an updating scheme whenever a concept drift is detected. The updating scheme that is presented here is a modification of a previously proposed methodology [28]. However, since local feature saliencies are added to the model and the number of hidden states is not fixed beforehand, the online scheme must be adapted.

At the beginning of any data-stream, an LFS-AsHMM with only one hidden state is learned from L instances and its BIC score per unit data is saved for the Page sequential test. Whenever a new instance arrives, it is concatenated to the current instances until ΔL instances are added. Then the log-likelihood per unit data of the full dataset is computed to be used in the Page test. Then, the Chernoff bounds are used to detect concept and feature drifts. If a novel concept drift appears, a new hidden state is added to the model and the SEM algorithm is executed only for the parameters related to the new added hidden state.

Suppose that the current model $\lambda = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ has N hidden states. Assume that at instance $\boldsymbol{x}^t$, a concept drift is detected and the model is updated. A new prior model $\lambda' = \{\boldsymbol{A}', \boldsymbol{B}', \boldsymbol{\pi}'\}$ is used to update the known information from the data stream.

To add a new hidden state in the transition matrix, the augmented matrix $\boldsymbol{C}$ is introduced:

$$\boldsymbol{C} = \left[\begin{array}{cccc|c} & & & & y_0 \\ & \boldsymbol{A} & & & \vdots \\ & & & & y_0 \\ \hline \frac{1}{N+1} & \cdots & \frac{1}{N+1} & & \frac{1}{N+1} \end{array}\right], \tag{26}$$

where y_0 must be a positive small number, in this study $y_0 = 1\times10^{-6}$. $\boldsymbol{C}$ will be used to generate a new prior transition matrix $\boldsymbol{A}'$ which enables the new model to determine the probabilistic transitions between the newly observed hidden state and the previously learned hidden states. We set $a'_{ij} = \frac{c_{ij}}{\sum_{j=1}^{N+1} c_{ij}}$, to ensure that $\boldsymbol{A}'$ is a Markov matrix.

For the prior values of $\boldsymbol{\eta}_{N+1,m}$, $\boldsymbol{\beta}_{N+1,m}$ and $\sigma^2_{N+1,m}$, we assume that the new hidden state in its relevant component can be represented as a naïve Bayesian network with no AR components . More precisely:

$$f_{N+1,m}(\boldsymbol{x}^t) = \mathcal{N}(\beta_{N+1,m}, \sigma^2_{N+1,m}), \quad \beta_{N+1,m} = \frac{1}{L}\sum_{i=1}^{L} x_m^{t-i},$$

$$\sigma^2_{N+1,m} = \frac{1}{L}\sum_{i=1}^{L}(x_m^{t-i} - \beta_{N+1,m})^2 \tag{27}$$

As explained in Section 4.1, a variable is noise to a process, if its parameters are fixed for the entire data stream. Therefore, we set the parameters ϵ_m and τ^2_m for all the data streams as follows:

$$\epsilon_m = \sum_{i=1}^{L}\frac{x_m^i}{L}, \quad \tau^2_m = \sum_{i=1}^{L}\frac{(x_m^i - \epsilon_m)^2}{L}. \tag{28}$$

These parameters are never updated. Finally, the parameter $\boldsymbol{\pi}'$ is updated as:

$$\boldsymbol{\pi}' = [\boldsymbol{\pi}|0]. \tag{29}$$

In other words, a concatenation operation with a zero to right. Once the prior model λ' is generated, the SEM algorithm is executed only on the parameters $\{A_{j,N+1}\}_{j=1}^N$, $\{A_{N+1,j}\}_{j=1}^N$, $\boldsymbol{\beta}_{N+1}$, $\boldsymbol{\eta}_{N+1}$ and σ^2_{N+1}. In this manner, the previously learned information is saved and used to identify recurrent states in data.

In many cases, the data streams may seem "infinite" and it is not possible to store all the captured data; hence a maximum window size L^* is imposed such that the learning and inference process does not require more than L^* instances. Once a new instance arrives and the buffer or data window of size L^* is already filled, the oldest instance is forgotten and the new instance is added to the processing data window.

5. Experiments

The experiments are focused on an online analysis, where the model must adapt itself when new trends in data appear and determine the feature relevancies for each time instance. We compare our strategy with the DFM-MCFS strategy proposed in [17] and the FSMCP strategy proposed in [18] (see Section 2). These methodologies are selected since they are the most recent feature selection in data streams in unsupervised problems. In the first case, the relevant features are updated whenever a buffer of data is filled. The methodology is compatible with any clustering model, in particular, we use the Gaussian mixture model, since it has been used previously in data stream problems [26]. In the second case, the features are updated when a test score is surpassed. In our model, the feature relevancies are updated as needed. We also compare ourselves against the model without the AR assumption (ablation study) to determine the relevance of having AR values. This model will be called LFS-BNHMM.

Regarding the novel concept hyperparameters used by the Page's sequential test and the Chernoff bounds must be fixed, in this study $e = 10^{-2}$, $h = 0.05$, $\varphi = 0.1$ and $c = 3$. With respect to the data window lengths, we set $L = 128$, $\Delta L = 10$ and $L^* = 4096$. Finally, $\tilde{\rho} = 0.9$. These parameters were obtained from a sensitivity analysis. Modifying these parameters may affect the ability of the methodology to detect novel concepts.

5.1. Synthetic data

5.1.1. Description

The synthetic data consist of $M = 10$ variables. Seven variables are relevant, two variables are noise, namely X_6 and X_9, and one variable X_3 is initially Gaussian noise but becomes relevant at certain time instances. We assume that the data contain $N = 5$ hidden states.

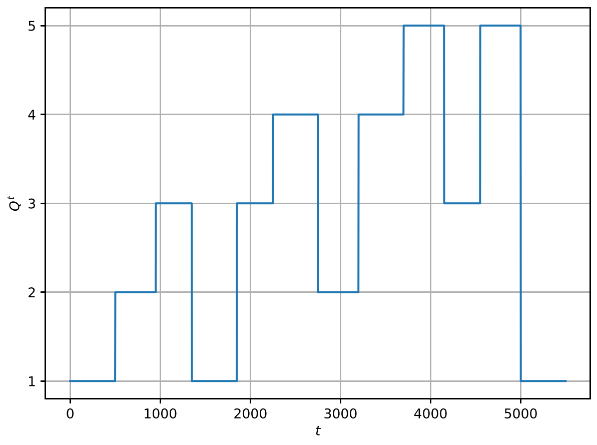


Fig. 3. Sequence of hidden states to generate synthetic data.

The mean and variance parameters are presented in the supplementary material. Note that, the dependencies of some variables change for every hidden state. Meanwhile, the parameters of the noise variables are the same for all the hidden states.

To simulate the data, a sequence of hidden states is needed. For this study, the sequence pictured in Fig. 3 is used. Two datasets are generated from this sequence, a time series for training and another for testing. All the models have an online training phase and then, their current parameters are tested. We then present the feature relevancies. In the case of our model, we also present the $g(i)$ evolution in the data stream and some of the learned context-specific Bayesian networks to gain insights.

5.1.2. Results

Fig. 4 shows the results in the testing phase of LFS-AsHMM. During the training data, six hidden states were discovered or inferred from the data. During the testing phase, one additional hidden state was found. The time where the hidden state was found is marked with a vertical dotted line. Note from the figure that variables X_6 and X_9 had a low relevancy as expected, since they are Gaussian noise during all the dynamic processes. Variable X_3 had some moments of high relevancy because its parameters differed from the noise level. The remaining variables showed an expected behaviour, since their relevancies were close to one most of the time. Nonetheless, it is relevant to observe that for some time instances, the relevancy of all variables reached close to zero. The reason behind this, is that the initial state (where all the variables are considered noise) was present in the data-stream. This implies that the noise component better represented the data and the relevancies in such cases had to be close to zero.

Table 3 shows the feature saliencies (ρ_{im}) during the training phase (with 6 hidden states) and the testing phase (where $Q = 7$ was added). Note that X_6 and X_9 were indeed noise variables, since for all the discovered hidden states, their relevancy was close to zero. Variable X_3 was irrelevant for most of the hidden states, but relevant in two of them $Q = 5, 6$. The remaining variables had a relevant values for most of the hidden states as expected.

We recall that it is assumed that at the first hidden state $Q = 1$, all the variables are noise, since no other concept is known. However, whenever a novel concept is discovered and the model is updated, the relevancy status of the features may change.

In Fig. 5(a), the results of DFM-MCFS are presented. It is observed that the relevancies had a greater variance than in the case of our proposed methodology. In particular, variables X_6 and X_9 had time periods where their relevancy was high, which is not true based on the construction of the dataset. Furthermore, relevant variables such as X_4 and X_{10} had a low relevancy for all the dynamic processes, which is

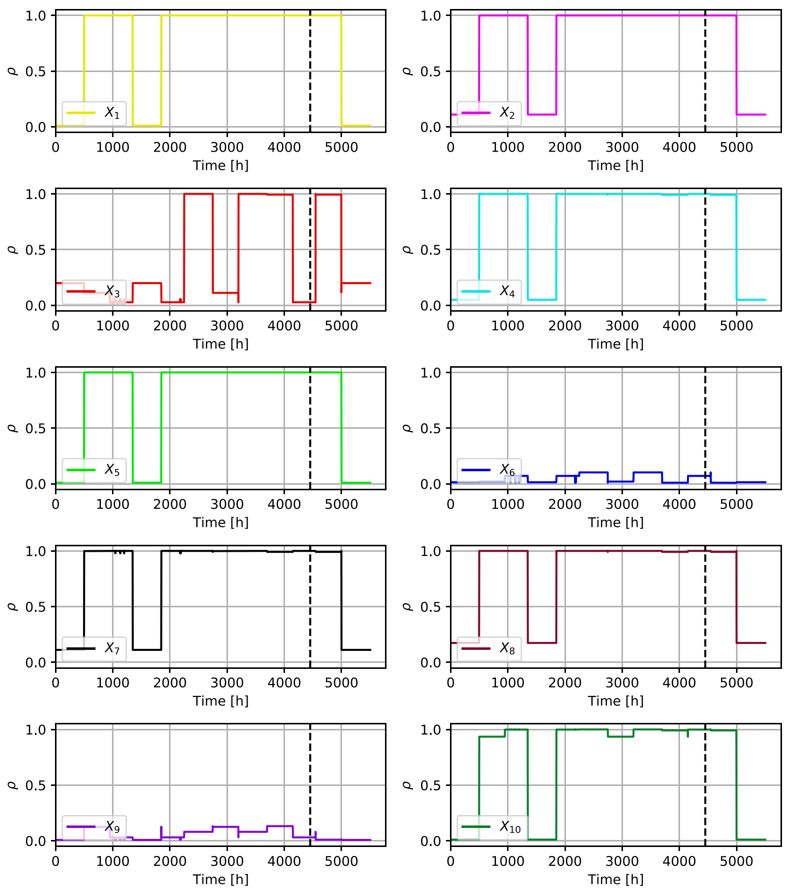


Fig. 4. Feature saliencies and drifts discovered during the testing phase by the proposed model LFS-AsHMM. Changes in feature relevancies are computed using the Viterbi algorithm.

Table 3
Feature saliencies discovered for all the discovered hidden states during the learning and testing phase.

$Q\backslash$ M	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
1	0.01	0.11	0.2	0.05	0.01	0.01	0.11	0.17	0.01	0.01
2	1.0	1.0	0.11	1.0	1.0	0.02	1.0	1.0	0.13	0.94
3	1.0	1.0	0.05	1.0	1.0	0.01	0.98	1.0	0.01	1.0
4	1.0	1.0	0.03	1.0	1.0	0.07	1.0	1.0	0.03	1.0
5	1.0	1.0	1.0	1.0	1.0	0.1	1.0	1.0	0.08	1.0
6	1.0	1.0	0.99	0.99	1.0	0.01	0.99	0.99	0.01	0.99
7	0.51	1.0	0.12	0.51	1.0	0.01	1.0	1.0	0.01	0.51

again, incorrect. Additionally, it is observed that the features changed their relevancy whenever the buffer of data was filled. In our case, the relevancy was unchanged unless an actual concept drift appeared in the data.

In Fig. 5(b), the results of FSMPC are pictured. It is observed that the relevancies had even higher variance over time. Relevant features such as X_4, X_5 and X_{10} had time instants where their relevancies were high as expected. However, there were also times where they were incorrectly low. Regarding the noisy variables X_6 and X_9, their relevancy level fluctuated greatly. There were time points where the value was one or close to one which is incorrect. Recall that this methodology updates the feature relevancies whenever a new instance arrives which explains the high variability. Nevertheless, the results were not as expected.

Fig. 6 shows the feature relevancies obtained by the model LFS-BNHMM. Five concept drifts were detected during the training dataset and three more were discovered in the testing data. When compared to the LFS-AsHMM, it is observed that this model is less robust to the data, and therefore, more states had to be learned to fully understand the data. Regarding the feature saliencies, variables X_6 and X_9 never surpassed the thresholds $\tilde{\rho} = 0.9$ and the remaining features were relevant. In this sense, the ablation is still powerful to provide feature saliencies information but it requires more hidden states when AR data is present in the data stream.

To compare the training and testing phases of the proposed algorithm, in Fig. 7 the BIC score per unit data is plotted for both the LFS-AsHMM and LFS-BNHMM in their testing and training phases. In (a), the BIC score per unit of data had time periods where it increased abruptly until a novel concept appeared and a model retraining was performed. In (c), the BIC score in the training also passed through time periods where it increased abruptly and concept drifts were detected. Nevertheless, it was also observed that some concept drifts were detected without a peak in the BIC, which caused that more hidden states were learned than in the case of the LFS-AsHMM. In (b), during the testing phase for the case of LFS-AsHMM, only one additional hidden state was learned since a monotonous increase in BIC was observed. In (d), in the case of the testing phase of LFS-BNHMM, three additional concepts drifts were learned and peaks in BIC were also observed. As before, this implies that the ablation of the model, (ignoring AR values) induces additional concept drifts, not only in the training phase, but also in the testing phase. Such events are highlighted with vertical dotted black lines. In the testing data, the BIC score per unit data was more stable and only one novel concept was found. This implies that the learned model in the training phase was not representative enough to explain all the testing phase.

Regarding the model inference, Fig. 8 includes the online clustering and its interpretation given by the Viterbi algorithm and the $g(i)$ function for the models LFS-AsHMM and LFS-BNHMMM. We observe

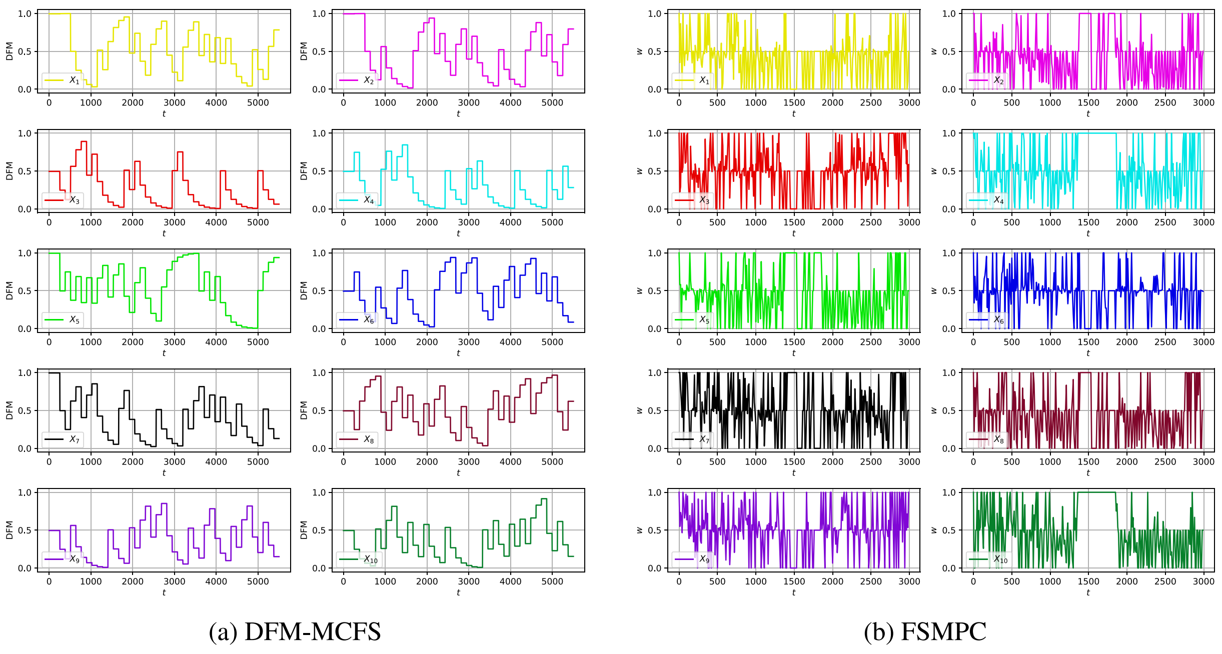


Fig. 5. Feature drifts discovered during the testing phase by (a) the DFM-MCFS and (b) FSMPC models.

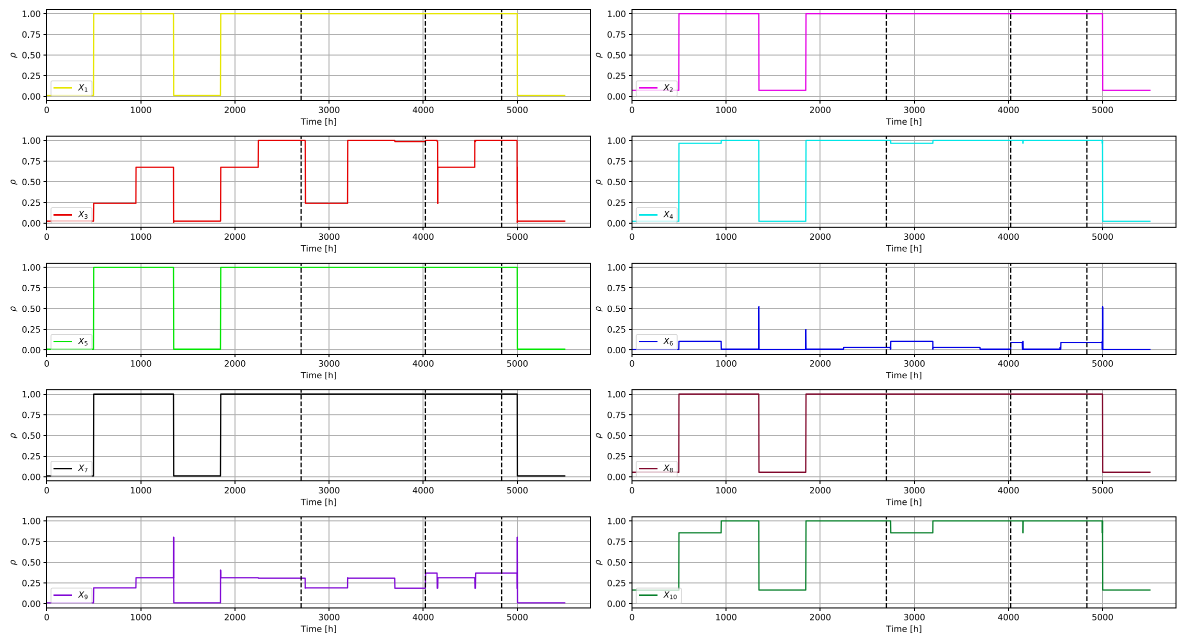


Fig. 6. Feature saliencies and drifts discovered during the testing phase by the proposed model LFS-BNHMM. The changes in feature relevancies are computed using the Viterbi algorithm.

that in (a), if the hidden state labelling path is compared to the real hidden state sequence plotted in Fig. 3, the prediction generally fits the ground truth well, with some outliers at the state transitions. In this sense, the model can be useful to learn and discover the intrinsic state and features drift in the data. On the other hand, in (b), the hidden state labelling path from LFS-BNHMM diverges from the real pattern observed in Fig. 3. This difference is specially clear at the time interval $[4500, 5000]$ where two peaks in the hidden state labelling should be seen, and such is not the case. The reason behind this, is due to the AR property which can describe the increase in magnitude of the relevant variables. Since LFS-BNHMM does not have the AR property, it is not capable of describing such increase in the data magnitude.

It must be recalled that we are not working with supervised data, and unsupervised models must discover patterns in data. Proof of this is the difference between the true number of hidden states (five from the data description) and the estimated number of hidden states in the LFS-AsHMM (seven in this case); however, the model could provide data insights that are helpful to understand and analyse the time series.

5.2. Real data

5.2.1. Ball-bearing degradation

Ball-bearings are relevant mechanical components inside industrial and nonindustrial machines. It is known that due to the application of mechanical loads such as high forces or temperatures,

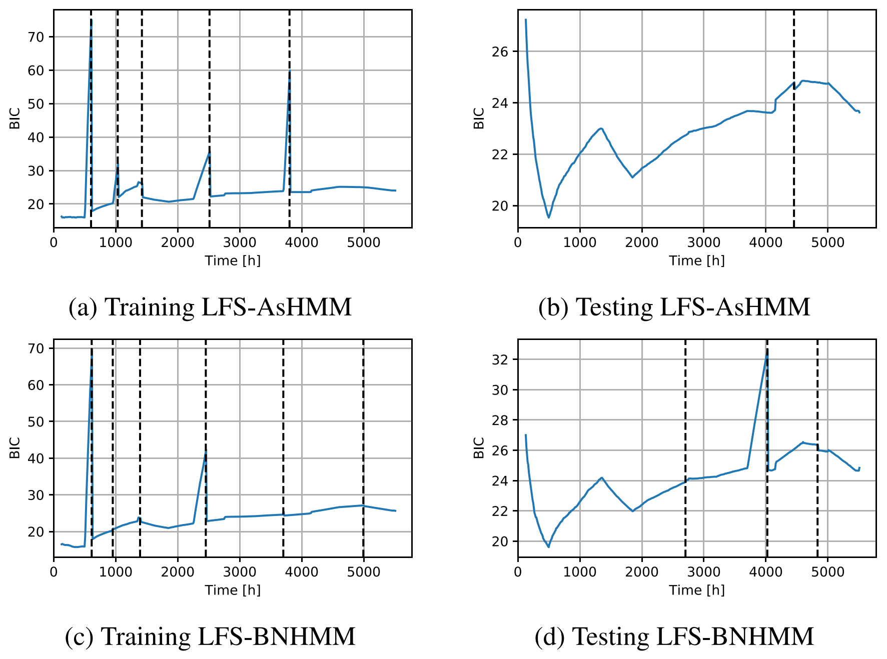


Fig. 7. BIC per unit data comparison between training and testing phases.

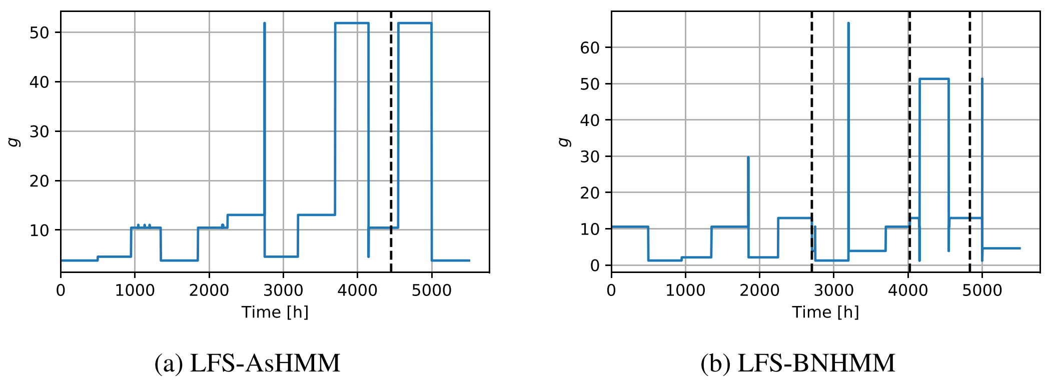


Fig. 8. Sequence of hidden states inferred during the testing phase based on the g function.

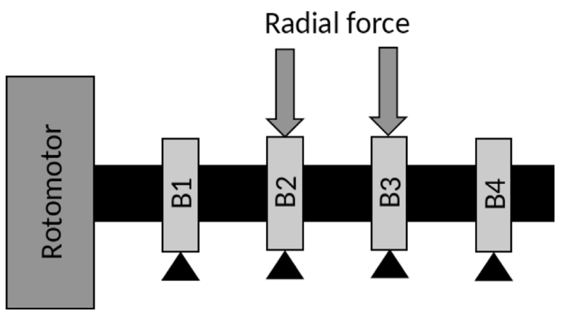
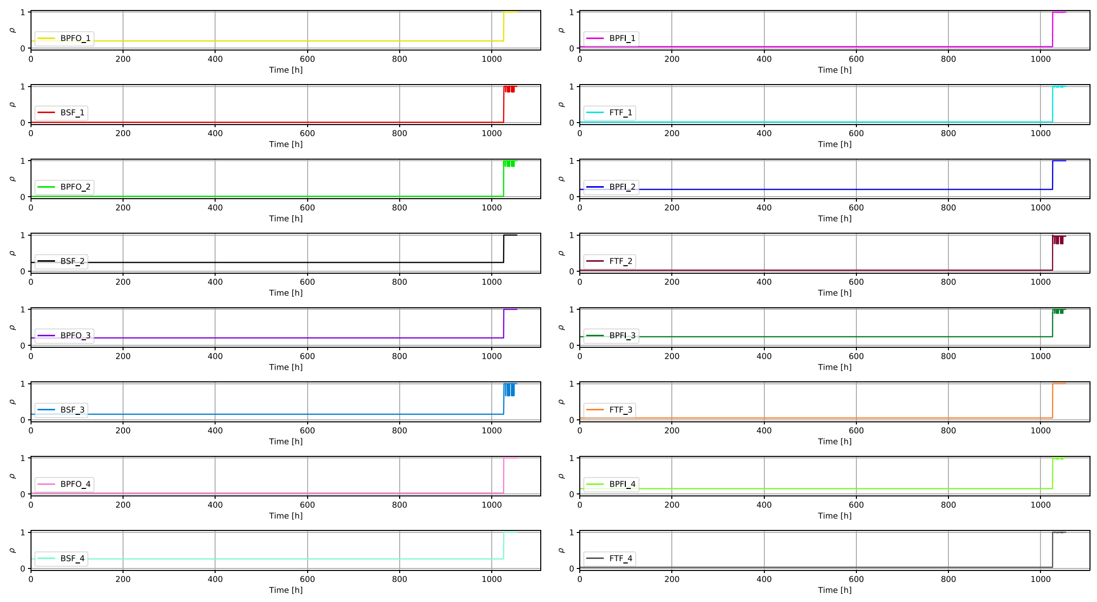


Fig. 9. Mechanical setup used in [29] to obtain data from run-to-failure ball-bearings.

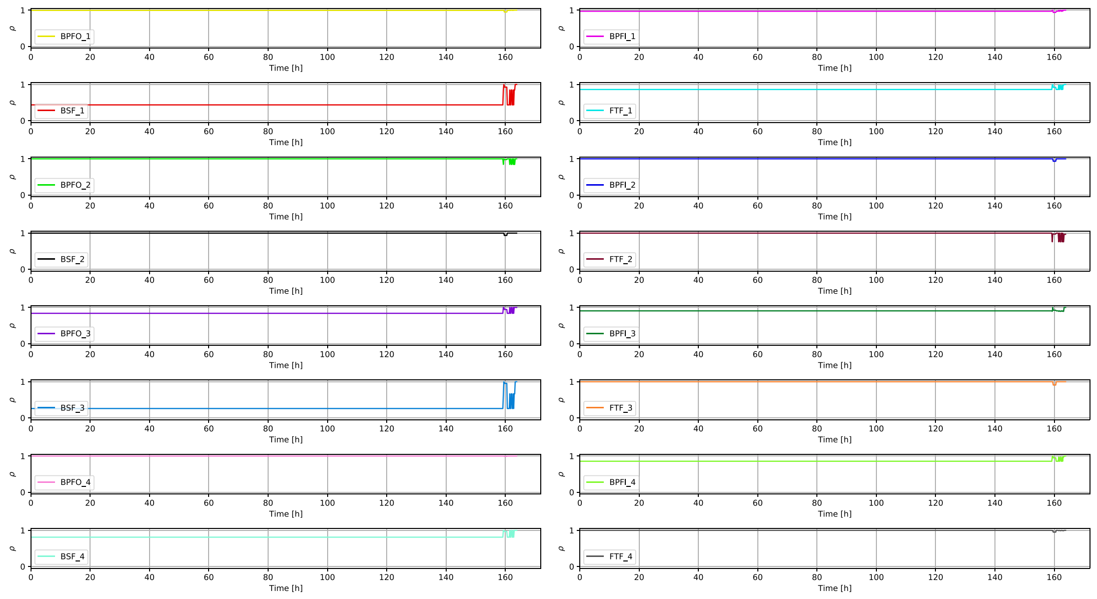
ball-bearings suffer from degradation and breakage. In industrial applications, the damage of one ball-bearing can compromise the productivity of manufacturing lines and induce losses in terms of time and money [30].

For this application we use the dataset provided by [29]. The machine-tool setup is shown in Fig. 9. It is known that the signals obtained from bearings using accelerometers contain noise that must be cleaned. In this article we use the methodology proposed by [31,32]. From the filtering process, we obtain the ball-bearing fundamental frequencies, namely, the ball pass frequency outer(BPFO) race, which is related to the ball-bearing outer race, ball pass frequency inner (BPFI) race, which is related to the ball-bearing inner race, ball spin frequency (BSF), which is related to the ball-bearing rollers and the fundamental train frequency (FTF), which is related to the ball-bearing cage, as features. It is known that the harmonics from these frequencies can provide information about the ball-bearing degradation process [28]. Additionally, it is known that greater the magnitudes of these frequencies is, the greater the degradation of the ball-bearing [30]. Nevertheless, in the early stages of a ball-bearing's life, harmonics and fundamental frequencies can be seen as noise and are irrelevant for ball-bearing health estimation. In this study, we aim to dynamically determine the features that are relevant and those that are not.

The dataset contains three run-to-failure time series of a mechanical setup. The mechanical setup consists of four ball-bearings, namely, B1,



(a) LFS-AsHMM test 1



(b) LFS-AsHMM test 2

Fig. 10. Feature drifts discovered during the testing phase by our model LFS-AsHMM.

B2, B3 and B4. Bearings B2 and B3 are under a controlled force. For more details about the mechanical setup, see [29]. Additionally, all the features were multiplied by 1000 to avoid numerical problems with small variances. For this application, we will focus on bearing B3, since it fails in two out of the three time series. We use the first (with respect to the acquisition date) two time series to train a model, which will be used to analyse the third time series. In the training signals, ball-bearing B3 failed with respect to its inner race. In the testing signal, ball-bearing B3 failed due to its outer race.

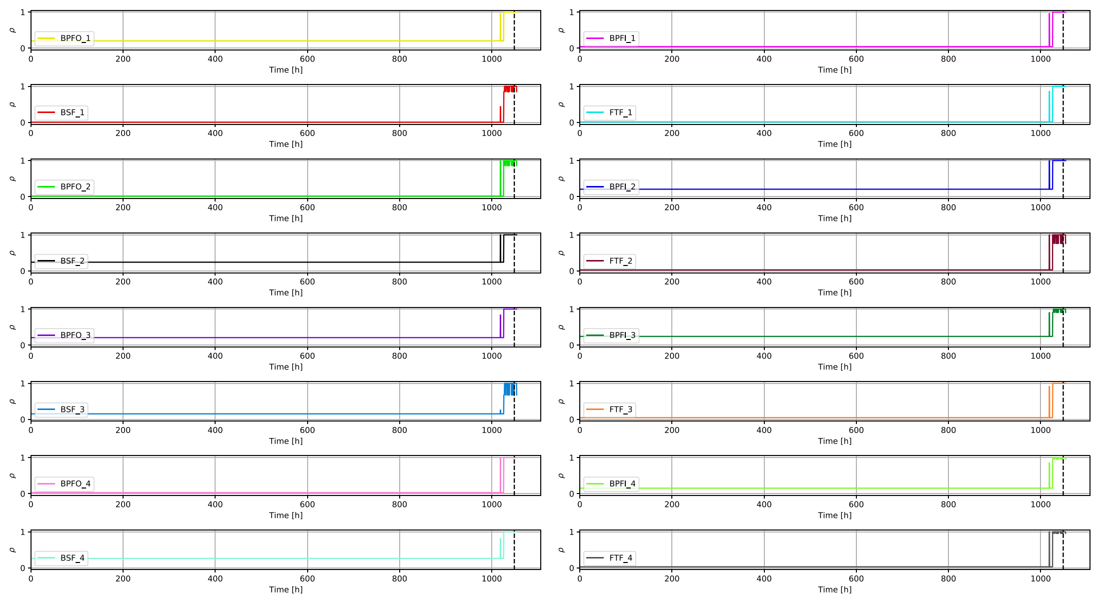
We use the fundamental frequencies and three harmonics as variables, hence sixteen features are used. The learned baseline model from the learning phase, will be used in the testing phase to detect feature drift. In the case of a novel concept due to an unknown concept or feature drift, it will be added to the model.

For the sake of robustness, we create a second training dataset consisting of the first and last time series and the second time series is used for testing. In this sense we cross-validate the obtained results, and see how the obtained results change.

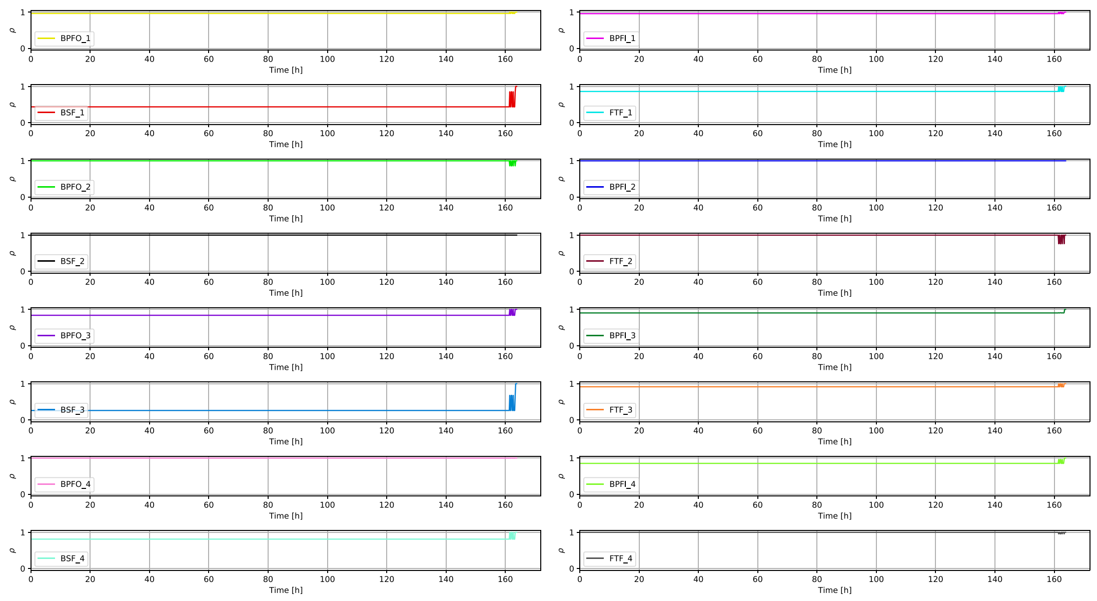
5.2.2. Ball-bearing results

Table 4 shows the obtained relevancies during the training phase when the first two learned signals were used for training. The variables with an index of 1, refer to the fundamental frequency; index i refers to the $i-1$ harmonic $i>1$. The most relevant result is that at the latest discovered trend, all the frequencies are relevant as expected. Nonetheless, for certain discovered trends, especially $Q=2$, it is observed that some variables do not surpass the threshold $\rho_{im}>\tilde{\rho}$, i.e., FTF_1 and BSF_1, $BPFO_3$, BSF_3, $BPFI_4$ and BSF_4. However, at the end of the process, the relevancies tended to increase with some exceptions.

In the online testing phase, the recorded evolution of relevancies is observed in Fig. 10 for testing signals for both models LFS-AsHMM and LFS-BNHMM. In (a), when LFS-AsHMM is used to validate the



(a) LFS-BNHMM test 1



(b) LFS-BNHMM test 2

Fig. 11. Feature drifts discovered during the testing phase by model LFS-BNHMM.

first testing signal, for most of the ball-bearing life, all the features behave as noise. Nonetheless, some hours before the ball-bearing breaks ($t = 1000$ h), all the features drifted to a relevant level, with some exceptions, where noisy drifts were observed, namely BSF_1, $BPFO_2$, BSF_3, FTF_2 and $BPFI_3$ and the relevancies oscillated between different levels. In spite of the noisy end behaviour, the model was capable of detecting the moment where the model was running to failure. It is noted that some relevancies were not used (see for example, the BSF_1 relevancies at $Q = 2$). This may indicate that in the training phase, more concepts were observed during the ball-bearing breaking process than in the testing phase. The same comments can be said about the ablation model LFS-BNHMM for the first testing signal, see Fig. 11 (a). In (b), most of the features are relevant for all the dynamic process, with some exceptions, e.g., BSF_1 and BSF_3, whose feature saliencies increase their values at the end of the ball-bearing life as in (a). This implies that the ball-bearing was already in a bad state. Evidence of this is that the ball-bearing in test 2 failed much earlier than in test 1. Additionally, it is remarkable that in the testing phase, for both tests, no novel trends were detected and the model never had to self-update. In this sense, the learned model could be sufficiently representative of both the testing data. The previous results also hold for the LFS-BNHMM model for the second testing signals as observed in Fig. 11 (b), which indicates that the AR components were not actually critical for this case study.

Fig. 12 shows the relevancy results for the other two methodologies for both testing signals. In (a), a plot of the periodic behaviours in all feature relevancies using the FDM-MCFS methodology is shown. However, none of them reached an important level of relevancy for all ball-bearing lives, as observed with our model. Nonetheless, the low relevancies were also latent during the ball-bearing failure times, which

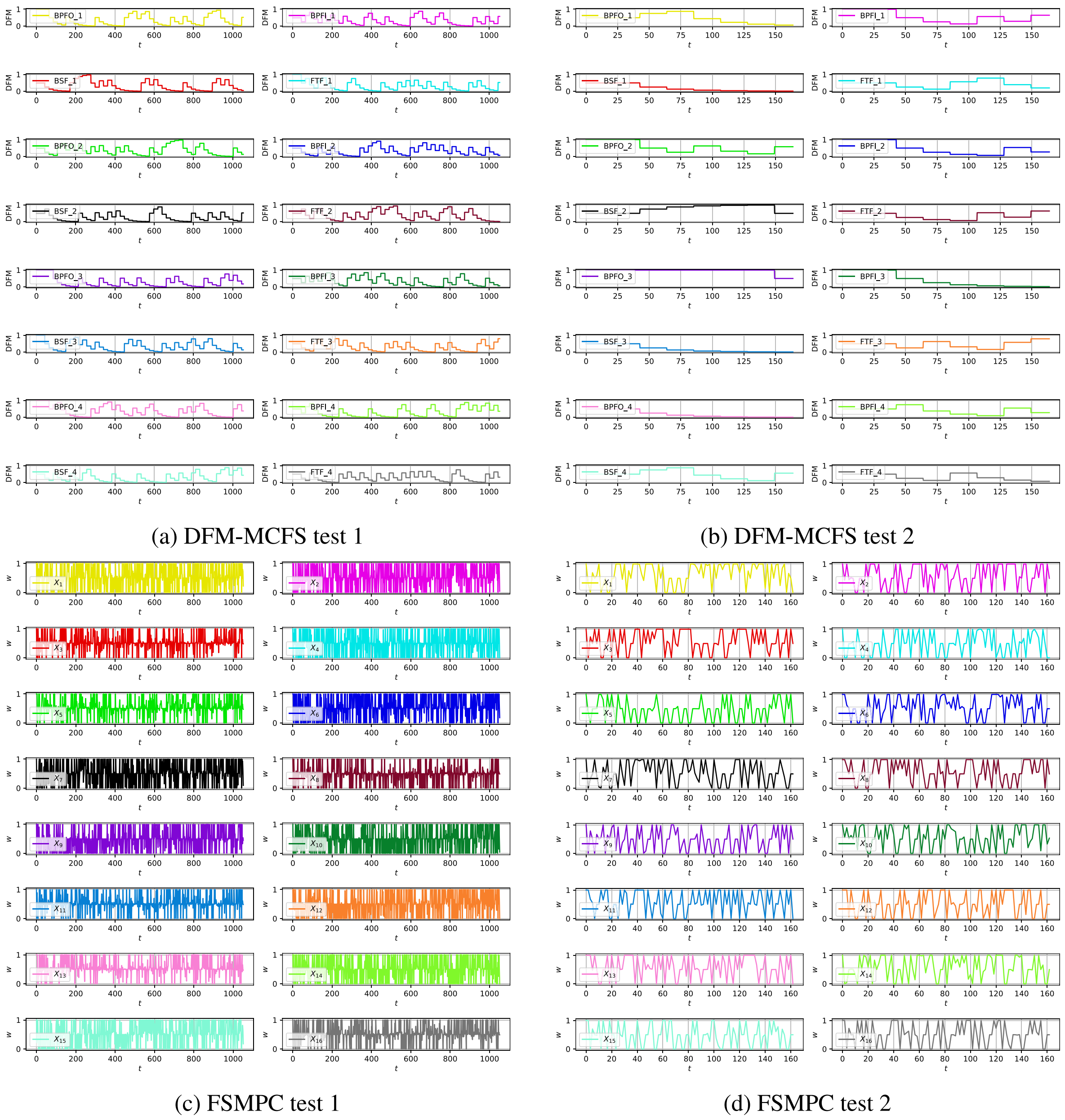


Fig. 12. Feature drifts discovered during the testing phase by (a) the DFM-MCFS and (b) FSMPC models.

Table 4
Feature relevancies ρ_{im} from the proposed model for the different frequency magnitudes found in the B3 learning and testing phases.

$Q\backslash$ m	$BPFO_1$	$BPFI_1$	BSF_1	FTF_1	$BPFO_2$	$BPFI_2$	BSF_2	FTF_2
1	0.2	0.04	0.01	0.02	0.02	0.21	0.24	0.03
2	0.99	0.97	0.44	0.86	1.0	1.0	1.0	1.0
3	1.0	0.94	1.0	0.66	0.99	1.0	1.0	1.0
4	1.0	1.0	0.85	0.98	0.85	1.0	1.0	0.76
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.97
$Q\backslash$ m	$BPFO_3$	$BPFI_3$	BSF_3	FTF_3	$BPFO_4$	$BPFI_4$	BSF_4	FTF_4
1	0.21	0.24	0.16	0.05	0.03	0.15	0.26	0.03
2	0.84	0.91	0.26	1.0	1.0	0.86	0.81	1.0
3	1.0	1.0	0.94	0.97	1.0	1.0	0.85	0.01
4	1.0	0.89	0.67	1.0	1.0	0.97	1.0	0.99
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

is not desired since the methodology could not provide insights into the ball-bearing degradation. Nevertheless, for the second testing signals in (b), the methodology actually provides some insights of the relevancy of the features. For instance BSF_1, $BPFO_1$, BSF_3 and FTF_4 decrease along time and other frequencies as $BPFO_3$ or BSF_2 remain relevant for all the ball-bearing life. But, it is expected from the ball-bearing literature, that the relevancy of the frequency components increase over time instead of decreasing, which in this case, does not hold. In (c) and (d), the FSMPC strategy shows a noisy behaviour for the entire life of the ball-bearing for both testing signals which is not helpful, since it did not provide any kind of feature insight or process understanding.

Fig. 13(a) shows the evolution of hidden states (Viterbi path with the g function) for the testing signal 1 by LFS-AsHMM. This sequence shows that the summed magnitudes of the ball-bearing were statistically constant during most of the ball-bearing life. However, as in the case of the feature relevancies, the magnitudes increase drastically in the ball-bearing final hours, indicating, as before, a worsening of the ball-bearing health. In the case of the second testing signal by LFS-AsHMM in (b), the model is also capable of detecting the change in magnitude at the end of the ball-bearings life, indicating the presence

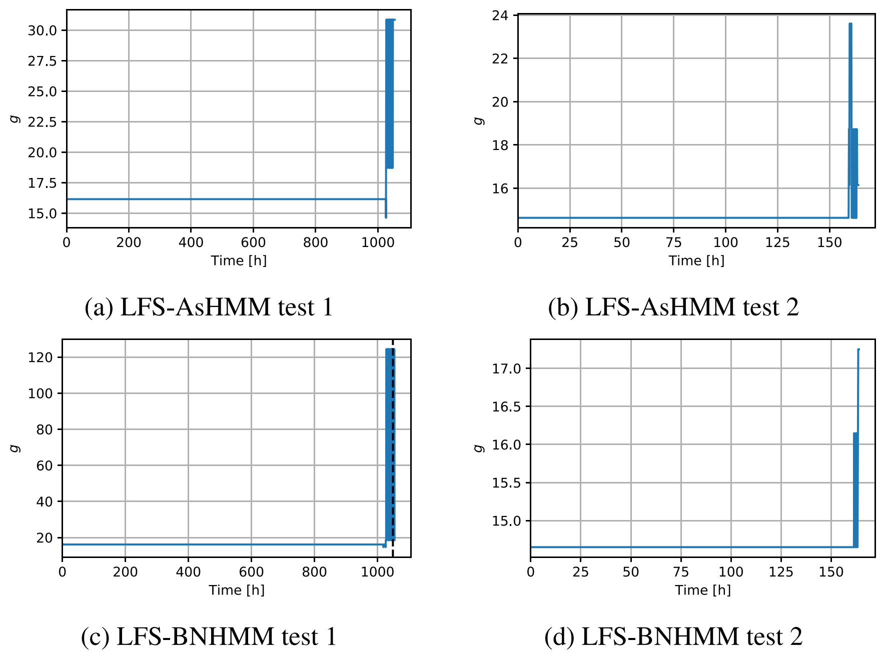


Fig. 13. Sequence of hidden states inferred by LFS-AsHMM during the testing phase of the ball-bearing.

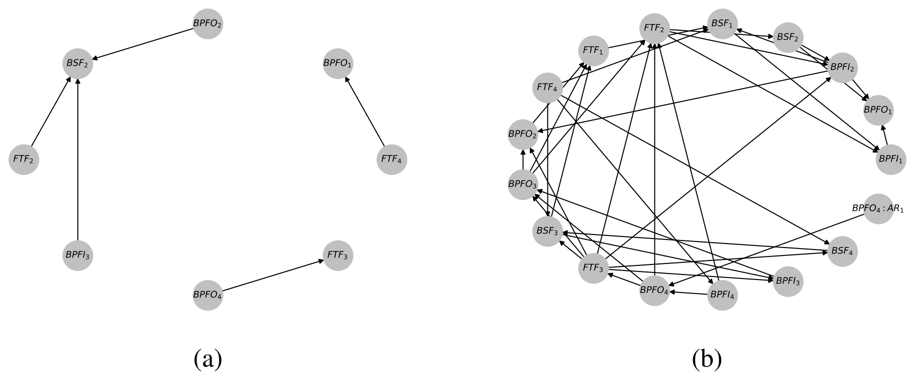


Fig. 14. Example of context-specific Bayesian networks learned by LFS-AsHMM from the ball-bearing degradation.

of a failure as in (a). Regarding the model LFS-BNHMM, it is observed that the same insights are extracted from both testing signals in (c) and (d). The model detected as well that at the end of the ball-bearing life, the magnitudes of the frequencies increase drastically indicating a soon failure.

Finally, as stated in the introduction, the model is capable of learning context-specific graphical models. These models can be used as an explanatory tool to understand and provide further data insights. Recall that the space of possible graphical models is determined by the features that fulfil the condition $\rho_{im} > \tilde{\rho}$. Fig. 14(a), is a graph when the ball-bearing was not in an advanced degradation state in the training phase. It is observed that the BPFO and FTF harmonic and fundamental frequencies are related. In other words, the behaviour of the ball-bearing outer ring and cage were dynamically dependent. Additionally, note that the BSF first harmonic was dependent on the harmonics of the BPFI, BPFO and FTF, which indicated that the ball-bearing rollers were dependent on the behaviour of the other ball-bearing components. The graph of Fig. 14(b) corresponds with a state where the ball-bearing was heavily damaged in the training phase. In this scenario, several dependencies between the ball-bearing frequencies are observed, which is evidence that the dynamical system was completely coupled and that all the ball-bearing parts were relevant for the degradation process. Additionally, an autoregressive behaviour in the third harmonic of the BPFO ($BPFO_4 : AR_1$) is observed, which means that the past values of the ball-bearing were also relevant to explain the current level of degradation.

In summary, the proposed methodology not only detects relevant features drift but also provides data insights. In addition, it is indeed capable of detecting the number of hidden states or concepts of a process, and it is capable of self updating in testing data if needed.

6. Conclusions

In this paper we proposed a new methodology to detect and learn feature drift in continuous unsupervised data. The proposed methodology uses a new embedded feature selection algorithm based on asymmetric hidden Markov models. The model has the property of “local feature relevancy”, which enables it to change the relevancy level of the features depending on the hidden states/concepts discovered in data. The methodology allows the model to self determine its number of hidden states as the data arrives in a data stream. Once a model is learned, it can be used in testing data to detect concept drift using the Viterbi algorithm. In each concept a set of relevant features is selected and a context-specific Bayesian network is used to explain the data. These context-specific Bayesian networks together with the localized feature relevancies, can provide a deeper data understanding of the data. Finally, a hidden state labelling equation is proposed to determine the magnitude of changes in concepts or hidden states taking into account the relevant and irrelevant features of each context.

The model was validated using synthetic and real data from degradation of ball-bearings. It was compared to other state of the art techniques that dynamically determine the relevance of features in unsupervised problems. It was observed that the most consistent results were obtained using the proposed methodology, which not only provides feature relevancies dynamically but also provides useful data insights.

In future work we want to extend the model to cases where the data are not necessarily Gaussian. In many cases and applications, when observing a data histogram, it is clear that it is not possible to represent the data with Gaussian distributions. In such cases, the data insights from the proposed model are biased and not representative. Additionally, in such cases is possible that the number of discovered hidden states tends to infinity which limits its use in real online applications.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Carlos Puerto-Santana reports financial support was provided by Spanish Ministry of Economy and Competitiveness. Carlos Puerto-Santana reports financial support was provided by Spanish Ministry of Science and Innovation. Carlos Puerto-Santana reports financial support was provided by H2020 IoTwins project.

Data availability

Data will be made available on request.

Acknowledgements

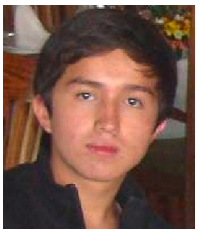
This study was partially supported by the Spanish Ministry of Economy and Competitiveness through the PID2019-109247GB-I00 project and by the Spanish Ministry of Science and Innovation through the RTC2019-006871-7 (DSTREAMS project). Additionally, by the H2020 IoTwins project (Distributed Digital Twins for industrial SMEs: a big-data platform) funded by the EU under the call ICT-11-2018-2019, Grant Agreement No. 857191.
00

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neucom.2023.126641.

References

[1] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[2] C. Villa-Blanco, C. Bielza, P. Larrañaga, Feature subset selection for data and feature streams: A review, Artif. Intell. Rev. (2022) 1–50.
[3] G. Oliveira, L.L. Minku, A.L.I. Oliveira, Tackling virtual and real concept drifts: An adaptive Gaussian Mixture Model Approach, IEEE Trans. Knowl. Data Eng. (2021) 1.
[4] J.P. Barddal, F. Enembreck, H.M. Gomes, A. Bifet, B. Pfahringer, Merit-guided dynamic feature selection filter for data streams, Expert Syst. Appl. 116 (2019) 227–242.
[5] T.M. Nguyen, Q.M.J. Wu, H. Zhang, Asymmetric mixture model with simultaneous feature selection and model detection, IEEE Trans. Neural Netw. Learn. Syst. 26 (2) (2015) 400–408.
[6] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1996, pp. 115–123.
[7] C. Puerto-Santana, P. Larrañaga, C. Bielza, Feature saliencies in asymmetric hidden Markov models, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1.
[8] M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, B. Thuraisingham, Classification and novel class detection of data streams in a dynamic feature space, in: Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, 2010, pp. 337–352.
[9] L. Wang, H. Shen, Improved data streams classification with fast unsupervised feature selection, in: 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2016, pp. 221–226.
[10] H.-L. Nguyen, Y.-K. Woon, W.-K. Ng, L. Wan, Heterogeneous ensemble for feature drifts in data streams, in: Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, 2012, pp. 1–12.
[11] S.C.H. Hoi, J. Wang, P. Zhao, R. Jin, Online feature selection for mining big data, in: Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, Association for Computing Machinery, 2012, pp. 93–100.
[12] H. Hoeltgebaum, N. Adams, C. Fernandes, Estimation, forecasting, and anomaly detection for nonstationary streams using adaptive estimation, IEEE Trans. Cybern. (2021) 1–12.
[13] W. Fan, N. Bouguila, An online approach for learning non-Gaussian mixture models with localized feature selection, in: 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2013, pp. 1–6.
[14] H. Huang, S. Yoo, S.P. Kasiviswanathan, Unsupervised feature selection on data streams, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Association for Computing Machinery, 2015, pp. 1031–1040.
[15] W. Shao, L. He, C.-T. Lu, X. Wei, P.S. Yu, Online unsupervised multi-view feature selection, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), 2016, pp. 1203–1208.
[16] X.-D. Wang, R.-C. Chen, F. Yan, H. Hendry, K-means clustering with feature selection for stream data, in: 2018 International Symposium on Computer, Consumer and Control (IS3C), 2018, pp. 453–456.
[17] C. Fahy, S. Yang, Dynamic feature selection for clustering high dimensional data streams, IEEE Access 7 (2019) 127128–127140.
[18] R. Ma, Y. Wang, L. Cheng, Feature selection on data stream via multi-cluster structure preservation, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Association for Computing Machinery, 2020, pp. 1065–1074.
[19] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, in: Readings in Speech Recognition, Morgan Kaufmann, 1990, pp. 267–296.
[20] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 1–38.
[21] J. Omura, On the Viterbi decoding algorithm, IEEE Trans. Inform. Theory 15 (1) (1969) 177–179.
[22] S. Adams, P. Beling, C. R., Feature selection for hidden Markov models and hidden semi-Markov models, IEEE Access 4 (2016) 1642–1657.
[23] C. Puerto-Santana, P. Larrañaga, C. Bielza, Autoregressive asymmetric linear Gaussian hidden Markov models, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2022) 4642–4658.
[24] J. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014).
[25] G. Schwarz, Estimating the dimension of a model, Ann. Statist. 6 (2) (1978) 461–464.
[26] J. Diaz-Rozo, C. Bielza, P. Larrañaga, Machine-tool condition monitoring with Gaussian mixture models-based dynamic probabilistic clustering, Eng. Appl. Artif. Intell. 89 (2020) 103434.
[27] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
[28] C. Puerto-Santana, P. Larrañaga, J. Diaz-Rozo, C. Bielza, An online feature selection methodology for ball-bearing harmonic frequencies based on HMMs, in: 16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021), Springer International Publishing, 2022, pp. 546–555.
[29] H. Qiu, J. Lee, J. Lin, G. Yu, Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics, J. Sound Vib. 289 (4) (2006) 1066–1090.
[30] P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. Puerto-Santana, C. Bielza, Industrial applications of machine learning, CRC Press, 2018.
[31] E. Bechhoefer, Envelope bearing analysis: Theory and practice, IEEE Aerosp. Conf. Proc. 2005 (2005) 3658–3666.
[32] Y. Wang, M. Liang, An adaptive SK technique and its application for fault detection of rolling element bearings, Mech. Syst. Signal Process. 25 (2010) 1750–1764.



Carlos Puerto-Santana received his bachelor's degree in Mathematics from Universidad de los Andes, Bogota, Colombia, in 2016 and his master's degree in Artificial Intelligence from Universidad Politécnica de Madrid, Spain, in 2018. He is currently a doctoral student at Universidad Politécnica de Madrid, Spain and researcher and developer in Aingura IIoT at San Sebastián, Spain.

Pedro Larrañaga received the M.Sc. degree in Mathematics (Statistics) from the University of Valladolid and the Ph.D. degree in Computer Science from the University of the Basque Country (excellence award). He has been a Full Professor in Computer Science and Artificial Intelligence with the Universidad Politécnica de Madrid (UPM), since 2007. Before moving to UPM, his academic career developed at the University of the Basque Country (UPV-EHU) at several faculty ranks: Assistant Professor, from 1985 to 1998,Associate Professor, from 1998 to 2004, and a Full Professor, from 2004 to 2007. He has published over 200 papers in high-impact factor journals, He has supervised over 30 Ph.D. theses. His research interests include the areas of probabilistic graphical models, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience, industry 4.0, and sports. He is Fellow of the European Association for Artificial Intelligence, since 2012, and a Fellow of the Academia Europaea, since 2018. He has received the 2013 Spanish National Prize in computer science and the prize of the Spanish Association for Artificial Intelligence in 2018 and the 2020 machine learning award of Amity University (India).

Concha Bielza received her M.S. degree in Mathematics from Universidad Complutense de Madrid, in 1989 and her Ph.D. degree in Computer Science from Universidad Politécnica de Madrid, in 1996 (extraordinary doctorate award). She is currently (since 2010) a Full Professor of Statistics and Operations Research with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid.

Her research interests are primarily in the areas of probabilistic graphical models, decision analysis, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience and industry. She has published more than 150 papers in impact factor journals and has supervised 20 Ph.D. theses. She was awarded the 2014 UPM Research Prize and the 2020 machine learning award of Amity University (India).