Online Active Learning for Human Activity Recognition from Sensory Data Streams

Saad Mohamad[a,b,*], Moamar Sayed-Mouchaweh[b], Abdelhamid Bouchachia[a]

[a] *Department of Computing, Bournemouth University, Poole, UK*
[b] *Department of Informatics and Automatics, Ecole des Mines, Douai, France*

Abstract

Human activity recognition (HAR) is highly relevant to many real-world domains like safety, security, and in particular healthcare. The current machine learning technology of HAR is highly human-dependent which makes it costly and unreliable in non-stationary environment. Existing HAR algorithms assume that training data is collected and annotated by human a prior to the training phase. Furthermore, the data is assumed to exhibit the true characteristics of the underlying distribution. In this paper, we propose a new autonomous approach that consists of novel algorithms. In particular, we adopt active learning (AL) strategy to selectively query the user/resident about the label of particular activities in order to improve the model accuracy. This strategy helps overcome the challenge of labelling sequential data with time dependency which is highly time-consuming and difficult. Because of the changes that may affect the way activities are performed, we regard sensor data as a stream and human activity learning as an online continuous process. In such process the leaner can adapt to changes, incorporate novel activities and discard obsolete ones. To this extent, we propose a novel semi-supervised classifier (OSC) that works together with a novel Bayesian stream-based active learning (BSAL). Because of the changes in the sensor layouts across different houses' settings, we use Conditional Re-

[*] Saad Mohamad is the corresponding author

Email addresses: smohamad@bournemouth.ac.uk (Saad Mohamad), moamar.sayed-mouchaweh@imt-lille-douai.fr (Moamar Sayed-Mouchaweh), abouchachia@bournemouth.ac.uk (Abdelhamid Bouchachia)

stricted Boltzmann Machine (CRBM) to handle the features engineering issue by learning the features regardless of the environment settings. CRBM is then applied to extract low-level features from unlabelled raw high-dimensional activity input. The resulting approach will then tackle the challenges of activity recognition using a three-module architecture composed of a feature extractor (CRBM), an online semi-supervised classifier (OSC) equipped with BSAL. CRBM-BSAL-OSC allows completely autonomous learning that adjusts to the environment setting, explores the changes and adapt to them. The paper provides the theoretical details of the proposed approach as well as an extensive empirical study to evaluate the performance of the approach.

Keywords: Activity Recognition, Data Streams, Active Learning, Online Learning.

1. Introduction

The recent advances of sensor technologies have led to affordable sensors with excellent performance, low weight, and low power consumption. In smart-homes, these sensors are widely deployed to collect data for monitoring purposes. From such data, useful knowledge can be extracted allowing for a variety of applications. In many of these applications, human activity recognition (HAR) is an essential task, such as health-care [1, 2], ambient assistive living [3, 4, 5, 6, 7] and surveillance-based security [8, 9, 10]. There are three main types of HAR, sensor-based [11, 12], vision-based [8] and radio-based [13]. Sensor-based methods rely on a large number of pervasive distributed sensors. Vision-based methods utilise image and video processing techniques to detect human activities. Radio-based methods use signal attenuation, propagation, and fading characteristics to detect human activities. In this paper, we are interested in sensor-based methods, which, unlike the other two classes of methods, do not work under a limited coverage area, enjoy the merits of information privacy, use widely available and affordable sensors and do not expose the human body to radiation that may raise health concerns.

A major contribution of this paper is the application of online and active learning for HAR. To the best of our knowledge, this study is the first in the field to propose online active learning to train an HAR algorithm. The HAR approaches have dominantly focused on traditional offline learning algorithms which assume that the training data is available prior to the training phase. Once this latter is exhausted, the learning algorithm is deployed and, cannot be trained any further even if performs poorly. This can happen if the used training data does not exhibit the true characteristics of the underlying distribution. In contrast, online learning views the data as a stream continuously arriving over time, where the learner can keep learning and adapt to changes. The standard assumption for activity recognition is that any new activities can be recognised using trained model on previously collected data. This is a strong assumption as it ignores natural changes in individuals' activity patterns and sensory measurements. In real-world situations, future data deviates from historical data because of changes in the activities induced by the resident. Such changes take place for several reasons: the way the people perform activities changes over time, their health conditions change, they perform novel activities, sensors displacement etc. We adopt an online learning algorithm to cope with these changes over time; hence, we view the sensory input as a continuous stream. Besides the aforementioned benefits, online learning copes well with memory and computation requirements because data samples are processed on-the-fly and then discarded immediately afterwards.

In the vast majority of HAR approaches, training data is assumed to be manually annotated (labelled). Such manual annotation is extremely hard and time-consuming task. For example, an expert will have to monitor the start and end of each of the resident's activity and attach labels to them. Furthermore, in the online setting, the data stream evolves, meaning that fresh labels are needed from time to time. Active learning (AL) is a paradigm of machine learning where the learning algorithm (learner) has control over the selection of training examples [14, 15]. AL deliberately queries particular instances to train the learner using as few labelled data instances as possible. In the context of

HAR, AL algorithm can query the user (individual carrying out the activities) about ambiguous or unknown activities in order to guide the learning process when needed. There exist two main approaches of AL: *pool-based selective sampling (PSS)* and *stream-based selective sampling (SSS)*. PSS is the most popular AL method, according to which the selection of instances is made by exhaustively searching in a large collection of unlabelled data gathered at once in a pool. Here, PSS evaluates and ranks the entire collection before selecting the best query. On the other hand, SSS scans through the data sequentially and makes query decisions individually. Different AL sampling criteria have been proposed [15]. Authors in [16] introduce one of the most general frameworks for measuring informativeness, *label uncertainty* sampling criterion, where the queried instances are those which the model is most uncertain about their label. Another popular AL sampling criterion framework is the *query-by-committee* [17]. Here, a committee of models trained on the same dataset are maintained. They represent different hypotheses. The data label about which they most disagree is queried. *Density-based* is another AL sampling criterion that differs from uncertainty and *query-by-committee* in that it uses unlabelled data for measuring the instance informativeness [18]. *Density-based* criterion assumes that the data instances in dense regions are more important.

In this paper, we propose a novel online semi-supervised classifier (OSC) equipped with an AL strategy. The proposed online classier is based on the Dirichlet process mixture model (DPMM) [19] with a stick-breaking prior [20] over the classes. Basically, the proposed model is a class-specific mixture model, where a mixture model is associated with each class. DPMM is a flexible non-parametric Baysian model which allows the complexity of the model to grow as more data is seen. Such a characteristic is useful in the case of data streams as not much prior knowledge is available. The application of stick-breaking prior over the classes allows accommodating new classes (activities). We employ a particle filter method [21, 22] to perform online inference.

We also propose a Bayesian stream-based AL strategy called (BSAL) that does not explicitly and purely adopt any of the aforementioned criteria. BSAL is an information theory based AL which aims at reducing the space of hypothesis by querying samples according to how much they are expected to reduce the model uncertainty [23]. On the contrary, decision theory based AL aims at reducing the prediction error by querying samples according to how much they are expected to reduce the future classification error [24]. While the two approaches seem quite distinct, they both aim at identifying data instances that give the largest reduction of the expected loss function. Thus, they mainly differ in the used type of loss functions. While decision theory based AL relies on the prediction error, information theory based AL losses involve the model parameters. One commonly used loss is the entropy of the model distribution.

Our proposed BSAL uses the Kullback-Leibler (KL) divergence as loss function (see Sec.3.2). We adopt an information based AL approach because it fits the Bayesian approach of the proposed semi-supervised classifier OSC (see Sec 3.1). Furthermore, there is no need to heuristically modify the loss function to account for the new classes because the loss involves the model distribution and not the prediction error. BSAL works completely online and is able to cope with the challenges associated with data streams, including the possible emergence of new classes.

As online learning copes well with the memory and computation requirements, OSC-BSAL can be used in mobile application which offers a handy way for the AL to query the annotators. The mobile device can apply OSC-BSAL on data streams coming from the mobile sensors or from other wearable and pervasive sensors. This example is a real world scenario where OSC-BSAL can be used. However in this paper, OSC-BSAL is evaluated on benchmark datasets. In our experimental setting, hundreds of sensors, wearable and distributed in the environment, are deployed resulting in high-dimensional data. Hence, designing hand-crafted features is extremely hard and time-consuming. The variation of the sensor network layouts in different homes makes the task even harder if portability of the system is desired. we tackle this issue by pre-trainning a Conditional Restricted Boltzmann Machine (CRBM) [25] to learn generic features from unlabelled raw high-dimensional sensory input. CRBM has been success-

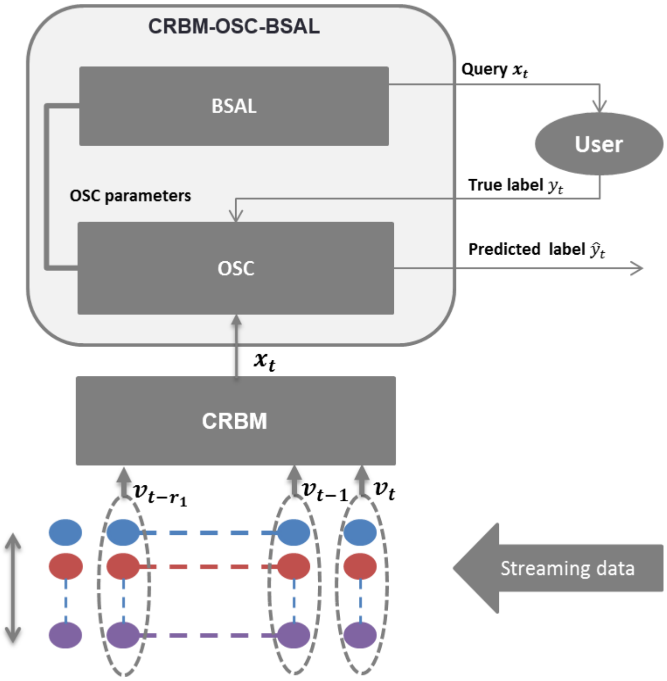


Figure. 1 General Architecture of CRBM-OSC-BSAL

fully applied in pattern recognition [9, 26, 27, 28, 29, 30, 31]. In this work, we apply CRBM to extract generic features from the sensory input. Details about CRBM, how it is trained and applied can be found in Appendix A. Figure 1 shows a simple sketch of the proposed architecture (CRBM-OSC-BSAL)

The rest of the paper is structured as follows: we discuss the related work and the motivation behind our work in Sec. 2. We describe the OSC-BSAL in Sec. 3. Empirical evaluation is presented in Sec. 4. Section. 5 concludes this paper.

2. Related Work and Motivation

Hand-crafted features have been the focus of most sensor-based HAR literature [32, 33], in which distinctive features are created or selected to train HAR systems. Statistical features such as mean, variance are utiliszed by [34, 35, 36, 37, 38, 11] as distinctive features of the sensory input. Such features are problem-specific and require the designer to understand the underlying problem to select and weight the most effective features using the expensive trial and error process. Any variation of the environment implies re-crafting the features, which is inadequate.

On the contrary, DL can be used to learn discriminative features from the data automatically and in a systematic way. DL learns different layers of features from low-level generic features to high-level features. DL has made a tremendous impact on different fields such as computer vision and natural language processing [39]. Recently, few studies have considered deep learning in sensor-based AR [40, 41, 42, 43, 44]. However, all of these studies work offline.

The great majority of HAR research is based on offline learning algorithms, in which the adaptation of the learner after the training is not possible. Recently, a few studies have targeted more challenging HAR settings where data comes in the form of streams [45].

The authors in [46] used online learning for HAR where the data is considered as a stream and where AL was applied to query activity labels when necessary. However, the proposed approach requires a labelled training set. In fact, there is two phases: (1) offline training phase and (2) online recognition and adaptation phase. In the offline phase, the model is built from a set of annotated sensory data that represents different activities. In the online phase, the recognition of unlabelled streaming data is performed. In this approach, the number of activities is assumed to be fixed and initialisation phase is required.

The authors in [47] proposed to address some of HAR challenges such data annotation through active learning. However, instead of using online learning to adapt the model when necessary, transfer learning is used. Models are trained on different collected houses and persons where transfer learning is employed to share the knowledge among these models. The authors also apply offline AL to obtain labels. However, any change is assumed to be represented in the training data, but on a large scale where different houses and different persons are covered. Hence, any change not occurring in the training data (e.g., emergence of novel activities) cannot be handled.

A similar architecture to ours is proposed in [48], where a hierarchical non-parametric Bayesian model is plugged on top of a deep network. The deep network learns low-level generic features, then the hierarchical non-parametric Bayesian model learns high-level features that capture correlations among low-level features. However, the model works offline, does not use AL and does not process time-series data.

The present paper goes beyond the sate-of-the-art methods by addressing the challenges of HAR in the smart-home setting where

1. CRBM allows to capture features that are less sensitive to the subtleties of sensory input. It learns generic features from the data in unsupervised way making CRBM-OSC-BSAL self-adjustable
2. OSC helps overcome dynamic changes within the same environment. It allows OSC-BSAL to be self-adaptive. OSC works online and adapts to change.
3. BSAL helps overcome hard and time-consuming activity annotation. It allows OSC-BSAL to be self-exploring. BSAL is the first AL that directly reduces the expected loss online, while considering the challenges of data streams.
4. The novel architecture combining CRBM, OSC and BSAL allows realistic HAR system with faster learning in comparison with other methods.

We review Dirichlet process (DP) in Appendix B. Being the core of OSC, DP is used as a non-parametric prior in Dirichlet process mixture model (DPMM) which, in contrast to the parametric prior, allows the number of components to vary during learning.

3. Online Semi-supervised Active Learning Classifier

In this section, we develop the proposed approach. We start by developing the online semi-supervised classier (OSC). Then, we move to the stream-based active learning algorithm (BSAL) which employs OSC.

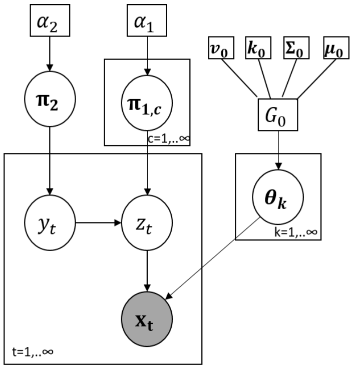


Figure. 2 Graphical model of OSC

3.1. ***Online Semi-supervised Classifier***

The proposed online semi-supervised classifier (OSC) can be expressed as a Dirichlet process mixture model (DPMM) with a new latent label variable y_t (observed after querying the sample). Figure 2 shows the structure of OSC in the form of a graphical model. $\boldsymbol{\pi_{1,c}}$ and $\boldsymbol{\pi_2}$ are drawn from stick-breaking processes $GEM(\alpha_1)$ and $GEM(\alpha_2)$ respectively; G_0 is a Normal-Inverse-Wishart distribution $NIW(.|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, k_0, v_0)$ with $\boldsymbol{\mu}_0$ is the prior of the clusters' means; $\boldsymbol{\Sigma}_0$ controls the variance among the means; k_0 scales the diffusion of the clusters' means and v_0 is the degree of freedom of the Inverse-Wishart distribution.

The label y_t is generated from a stick-breaking prior. While z_t selects the component generating x_t, y_t selects the stick-breaking component generating z_t. Label y_t selects from different mixture models associated with different classes. The model is updated as follows. If the data samples received are unlabelled, the whole model is updated such that the class variable is marginalized out. Otherwise, only the mixture model associated with the same class as the sample is updated with the new data sample. A particle filter method derived from [21, 22] is used to perform the online inference.

Following [21], we introduce a state vector H_t that summarizes the data seen up to time t. Hence, $H_t = \{z_t, m_t, \boldsymbol{n}_t, \boldsymbol{s}_t\}$ can replace all the statistics used in OSC, where m_t is the number of components; $\boldsymbol{n}_t$ is a matrix with rows referring to the number of data samples labeled to the existing classes and columns referring to the number of data samples assigned to the existing components; and $\boldsymbol{s}_t$ is the sufficient statistics for all mixture components(i.e., means and scatter matrices). The notations used in this section are summarized in Appendix C.

Every time a new sample is received, OSC carries out three steps: prediction, updating and re-sampling. While updating step differs according to whether the data sample is labelled or not, prediction and re-sampling steps are applied in the same way for both cases. The 3 steps of OSC are described as follows:

3.1.1. ***Prediction:***

Given the concentration parameters α_1, α_2 and the prior distribution parameters $\{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, k_0, v_0\}$, we aim at computing the conditional probability of the label given a data sample, $p(y_t|\boldsymbol{x}_t, D_{t-1})$:

$$p(y_t|\boldsymbol{x}_t, D_{t-1}) \propto p(\boldsymbol{x}_t|y_t, D_{t-1})p(y_t|D_{t-1}) \tag{1}$$

where D_{t-1} represents all data samples previously seen along with their labels if provided. Details on how Eq. (1) is computed can be found in Appendix D.

The solution of Eq. 1 depends only on the elements of the state vector along with its posterior distribution (see Appendix D). Thus, we need to track this posterior online by approximating it with a set of P particles. Upon the arrival of a new data point, the particles are extended to include a new assignment z_t assuming that the previous assignments are known and fixed. Thus, the task is to update the posterior of the extended particles at time t, $p(H_t|D_t)$, given that the posterior at $t-1$, $p(H_{t-1}|D_{t-1})$ is known. In order to prevent combinatorial explosion, we use the re-sampling technique proposed in [22] which retains only P particles. Therefore, we approximate the posterior at time t using the following updating and re-sampling steps:

3.1.2. Updating:

$$p(H_t|D_t) \propto \int_{H_{t-1}} p(H_t|H_{t-1}, y_t, \boldsymbol{x}_t) p(y_t, \boldsymbol{x}_t|H_{t-1}) p(H_{t-1}|D_{t-1}) \qquad (2)$$

Given the P particles along with their weights, $p(H_{t-1}|D_{t-1}) = \sum_{i=1}^{P} w_{t-1}^{(i)} \delta(H_{t-1} - H_{t-1}^{(i)})$, the update can be written as follow.

$$p(H_t|D_t) \propto \sum_{i=1}^{P} p(H_t|H_{t-1}^{(i)}, y_t, \boldsymbol{x}_t) p(y_t, \boldsymbol{x}_t|H_{t-1}^{(i)}) w_{t-1}^{(i)} \qquad (3)$$

The solution of the second term of Eq. (3) can be found in Appendix D (Eq. (D.3) and Eq. (D.2)). Following the updating step, the number of resulting particles for each $H_{t-1}^{(i)}$ becomes equal to the number of existing components $m_{t-1}^{(i)} + 1$. The new assignments z_t lead to different configurations of the new particles. Therefore,

$$\begin{aligned} p(H_t^{(j)}|H_{t-1}^{(i)}, y_t, \boldsymbol{x}_t) &= p(z_t = j_1|H_{t-1}^{(i)}, y_t, \boldsymbol{x}_t) \\ &\propto p(\boldsymbol{x}_t|y_t, z_t = j_1, H_{t-1}^{(i)}) p(z_t = j_1|y_t, H_{t-1}^{(i)}) \qquad (4) \end{aligned}$$

where $j = f(i, j_1)$ and $f(a, b) = \frac{1}{2}(a + b)(a + b + 1) + b$ is the Cantor pairing function which uniquely encodes the assignment j_1 and the particle number i into a single natural number. By solving Eq. (4), we determine the weight of the new particle $w_t^{(j)}$. Equation 4 is computed in Appendix D (the first term of Eq. (4) is computed in Eq. (D.6), and the second term is computed in Eq. (D.8)).

The elements of new state vector $H_t^{(j)}$ are updated as follows:

$$H_t^{(j)} = \left\{ \begin{array}{lll} z_t^{(j)} & = j_1 & j_1 \text{ is an existing component} \\ n_{y_t,j_1,t}^{(j)} & = \lambda n_{y_t,j_1,t-1}^{(i)} + 1 & \\ n_{h,k,t}^{(j)} & = \lambda n_{h,k,t-1}^{(i)} \quad \forall h \neq y_t, \forall k \neq j_1, h \in C_t, k \leq m_t^{(i)} & \\ \boldsymbol{su}_{j_1,t}^{(j)} & = \frac{\lambda n_{:,j_1,t-1}^{(i)} \boldsymbol{su}_{j_1,t-1}^{(i)} + \boldsymbol{x}_t}{n_{:,j_1,t}^{(i)}} & \\ \boldsymbol{sc}_{j_1,t}^{(j)} & = \lambda \boldsymbol{sc}_{j_1,t-1}^{(i)} + n_{:,j_1,t-1}^{(i)} \boldsymbol{su}_{j_1,t-1}^{(i)} \boldsymbol{su}_{j_1,t-1}^{(i)T} & \\ & -n_{:,j_1,t}^{(i)} \boldsymbol{su}_{j_1,t}^{(i)} \boldsymbol{su}_{j_1,t}^{(i)T} + \boldsymbol{x}_t \boldsymbol{x}_t^T & \\ & & \\ z_t^{(j)} & = m_{t-1}^{(i)} + 1 & j_1 \text{ is a new component} \\ m_t^{(j)} & = m_{t-1}^{(i)} + 1 & \\ n_{y_t,j_1,t}^{(j)} & = 1 & \\ n_{h,k,t}^{(j)} & = \lambda n_{h,k,t-1}^{(i)} \quad \forall h \neq y_t, \forall k \leq m_{t-1}^{(i)}, h \in C_t & \\ \boldsymbol{su}_{j_1,t}^{(j)} & = \boldsymbol{x}_t & \\ \boldsymbol{sc}_{j_1,t}^{(j)} & = \boldsymbol{0} & \end{array} \right. \tag{5}$$

where λ is a memory factor which allows the components to adapt with change, C_t is the set of the labels of all existing classes. If the label y_t is unknown, we consider it as a latent variable. The posterior $p(H_t|D_t)$ in the update Eq. (3) can be written as follows:

$$p(H_t|D_t) = \sum_{y_t} p(H_t|D_{t-1}, \boldsymbol{x}_t, y_t) p(y_t|D_{t-1}, \boldsymbol{x}_t) \tag{6}$$

$$p(y_t|D_{t-1}, \boldsymbol{x}_t) \propto p(\boldsymbol{x}_t|y_t, D_{t-1}) p(y_t|D_{t-1}) \tag{7}$$

Equation 7 is computed in Appendix D (the first and second terms of Eq. (7) are computed in Eq. (D.3) and Eq. (D.2) respectively). The first term of Eq. (6) is already computed in Eq. (3), where the label is assumed to be known. We can see from Eq. (6) that for each new assignment z_t, there is a mixture of particles that depends on j and the different labels y_t. We re-define the state vector H_t to accommodate y_t as hidden variable when it is unknown. Thus, $H_t' = \{H_t, y_t\}$, where the different particles are determined now by both z_t and y_t,

$$\begin{aligned} H_t'^{(j)} &= \{H_t^{(j_2)}, y_t = j_3\} \\ j &= f(j_2, j_3) \end{aligned} \tag{8}$$

where j_3 can be either an existing or a new class. To compute the weight and the update of the new state vector, we follow the same trend as in Eq. (3), Eq. (4) and Eq. (5).

$$p(H_t'|D_t) \propto \sum_{i=1}^{P} p(H_t'|H_{t-1}'^{(i)}, \boldsymbol{x}_t) p(\boldsymbol{x}_t|H_{t-1}'^{(i)}) w_{t-1}'^{(i)} \tag{9}$$

$$\begin{aligned} p(H_t'^{(j)}|H_{t-1}'^{(i)}, \boldsymbol{x}_t) &= p(z_t = j_1, y_t = j_3|H_{t-1}'^{(i)}, \boldsymbol{x}_t) \\ &\propto p(\boldsymbol{x}_t|y_t = j_3, z_t = j_1, H_{t-1}'^{(i)}) p(z_t = j_1, y_t = j_3|H_{t-1}'^{(i)}) \end{aligned} \tag{10}$$

After updating the P particles with all the possible new assignments z_t and y_t, we end up with M combinations of the P particles with the new assignments, along with the weights $w_t'^{(j)}$. So, we move to the next step which reduces the number of created particles to a fixed number P.

3.1.3. ***Re-sampling:***

We follow the resampling technique proposed in [22] which discourages the less likely particles (configurations), and improves the particles explaining the data better. It keeps the particles whose weight is greater than $1/\kappa$, and re-samples from the remaining particles. The variable κ is the solution of the following equation:

$$\sum_{j=1}^{M} \min\{\kappa w_t'^{(j)}, 1\} = P \tag{11}$$

The weight of re-sampled particles is set to $1/\kappa$, and the weight of the particles greater than $1/\kappa$ is kept unchanged. All the re-sampled particles are ensured to be re-sampled once.

To illustrate how the Re-sampling work, consider the following example. Set $P = 3$, $m = 5$ and the wrights to be $\{0.4, 0.2, 0.1, 0.3, 0\}$. By solving Eq. (11), we get $\kappa = 10/3$. Thus, the samples with the weights $\{0.4, 0.3\}$ are maintained and we re-sample one particle from the remaining $\{0.2, 0.1, 0\}$

3.2. ***Active Learning Approach***

We propose an AL algorithm that deliberately queries particular instances to train OSC using as few labelled data instances as possible. In the context of HAR, the labels are the human activities, and the AL algorithm queries the user (individual carrying out the activities) about some activities. Thus, extensive queries will be annoying and must be avoided. The notations used in this section are summarised in Appendix C. Note that Although the proposed AL is independent of the online classification model, OSC's use of non-parametric Bayesian prior over classes allows the AL to handle the *concept evolution* problem.

Most AL approaches are basically derived from the general approach of finding queries that result of the largest reduction in the expected loss. Let Ω denote the set of variables that can be fully random or include some random elements. Let L refer to the loss function. The expected loss function that AL aims to reduce can be expressed as follows:

$$R = E_{\Omega}[L(\hat{\Omega}, \Omega)] \tag{12}$$

where the hat over Ω (ie., $\hat{\Omega}$) refers to an estimated set of variables. Depending on the loss functions involved, AL can be divided to two main groups: information-based and decision-based AL. If we take Ω as the set of vectors consisting of observed set X and latent set Y elements, we can end up with the expected classification error:

$$R = \int_{\boldsymbol{x}} L(\hat{p}(y|\boldsymbol{x}), p(y|\boldsymbol{x}))p(\boldsymbol{x})d\boldsymbol{x} \tag{13}$$

Here, the loss function involves the prediction error. In AL, we request information about certain samples. That is the learner queries the latent labels Y of some subset of $X \subset \Omega$. We introduce a binary set Q whose elements are attached to the vectors in the set Ω. If an element $q \in Q = 1$, the latent elements of the corresponding vector in Ω are queried. Equation (13) can be seen as the core of most heuristic and non-heuristic decision-based AL. Many active learning approaches seek to minimize an approximation of the expected error of the learner Eq. (13) [24, 49, 50]. In our previous work [51, 52], we proposed an AL strategy that seeks to minimize Eq. (13) online, while considering the challenges of data streams.

Assuming that there is a model generating the data. If Ω is taken to be the set of the true model parameters $\boldsymbol{\psi}$, then the loss is over the model parameters. We obtain the following risk function:

$$R = E_{\boldsymbol{\psi}}[L(\hat{\boldsymbol{\psi}}, \boldsymbol{\psi})] \tag{14}$$

Similar to Eq. (13), Eq. (14) can be seen as the core of most heuristic and non-heuristic information-based AL. The authors in [53, 54] use the entropy of the model as the loss function. In this paper, we propose a Bayesian stream-based AL (BSAL) inspired from information-based AL. BSAL is designed to cope with the challenges of data streams (infinite length, evolving nature, emergence of new classes). Information-based AL fits better the nature of the proposed Bayesian semi-supervised classifier (OSC), where the uncertainty over the model parameters is systematically expressed. Because information-based loss functions are over the model distribution, BSAL can easily deal with the challenges of data streams. In fact, the task is only to take the decision whether to online query or not, while OSC works online and accommodates novel classes. Thus, BSAL is completely compatible with OSC.

In the following, we discuss the offline AL strategy to minimize an approximation of Eq. (14), then we present our online AL strategy. The authors in [23] propose an algorithm that selects queries in a greedy way in order to improve the model accuracy as much as possible. Their original goal is to minimize the loss $L(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}})$ incurred by using a single representation $\hat{\boldsymbol{\psi}}$ of the model instead of the true model parameters $\boldsymbol{\psi}$. As the true model parameters are unknown, authors estimate them using the posterior of the Bayesian model parameters $p(\boldsymbol{\psi}|X, Y)$. Therefore, the risk associated with a particular $\hat{\boldsymbol{\psi}}$ with respect to $p(\boldsymbol{\psi}|X, Y)$ can be expressed as follows:

$$R(p(\boldsymbol{\psi}|X, Y), \hat{\boldsymbol{\psi}}) = E_{\boldsymbol{\psi} \sim p(\boldsymbol{\psi}|X,Y)}[L(\boldsymbol{\psi}, \hat{\boldsymbol{\psi}})] \tag{15}$$

The point estimate $\hat{\boldsymbol{\psi}}$ that minimizes the risk is defined as the Bayesian point estimate. By fixing $\hat{\boldsymbol{\psi}}$ to the Bayesian point estimate, the resulting risk depends only on the model posterior. The authors in [23] adopted a pool-based AL approach, where the samples that reduce the risk the most are selected. Because the labels are unknown a priori, the expected risk, given the query, is computed as follows:

$$\hat{R}\big(p(\boldsymbol{\psi}|X, Y), \hat{\boldsymbol{\psi}}; Q = \boldsymbol{q}\big) = E_{Y_{S(Q)} \sim p(Y_{S(Q)}|X)}[R(p(\boldsymbol{\psi}|X, Y_{S(Q)}), \hat{\boldsymbol{\psi}})] \tag{16}$$

where clamping $\boldsymbol{q}$ to Q refers to the state of querying samples that correspond to the elements in $\boldsymbol{q}$ which are equal to one. The function $S(Q)$ returns the indices of the elements in Q that are equal to 1 (i.e., the samples to be queried). If $S(Q)$ is empty, the expected risk in Eq. (16) becomes:

$$\hat{R}\big(p(\boldsymbol{\psi}|X, Y), \hat{\boldsymbol{\psi}}; Q = \boldsymbol{q}\big) = R(p(\boldsymbol{\psi}|X), \hat{\boldsymbol{\psi}}) \tag{17}$$

The goal of AL is to query the data samples that result in maximizing the difference between the risk (Eq. (15)) and the expected risk (Eq. (16)).

$$\hat{\Delta}(X, Y|\boldsymbol{q}) = R(p(\boldsymbol{\psi}|X), \hat{\boldsymbol{\psi}}) - \hat{R}(p(\boldsymbol{\psi}|X, Y), \hat{\boldsymbol{\psi}}; Q = \boldsymbol{q}) \tag{18}$$

Given a stream of samples, BSAL evaluates each sample at time t before discarding it. Here, the querying random variable refers to whether or not the input sample at time t is queried. The Data X and Y become the data seen so far $\{D_{t-1}, x_t, y_t$. Since in online active learning, we are sure that at time t no samples within the data seen up to time $t-1$ D_{t-1} will be queried, we condition on D_{t-1}. Therefore, Equation (18) can be reformulated as follows:

$$\hat{\Delta}(\boldsymbol{x}_t, y_t|D_{t-1}, q_t) = R(p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t), \hat{\boldsymbol{\psi}}_t) - \hat{R}(p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t, y_t), \hat{\boldsymbol{\psi}}_t; q_t) \tag{19}$$

where:

$$R(p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t), \hat{\boldsymbol{\psi}}_t) = E_{\boldsymbol{\psi}_t \sim p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t)}[L(\boldsymbol{\psi}_t, \hat{\boldsymbol{\psi}}_t)] \tag{20}$$

If the sample at time t is not queried ($q_t = 0$), the current expected risk (second term of Eq. (19)) is equal to the current risk (first term of Eq. (19)). Otherwise, the current expected risk can be written as follows:

$$\hat{R}(p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t, y_t), \hat{\boldsymbol{\psi}}_t; q_t = 1) = E_{y_t \sim p(y_t|D_{t-1}, \boldsymbol{x}_t)}[R(p(\boldsymbol{\psi}_t|D_{t-1}, \boldsymbol{x}_t, y_t), \hat{\boldsymbol{\psi}}_t)] \tag{21}$$

Intuitively, Eq. (19) measure the discrepancy between the risk, representing the model uncertainty computed from the data seen so far and the expected risk with respect to y_t. This expresses how much querying y_t influence the model uncertainty. Note that the DP prior used in OSC over the classes allows posing distribution over novel classes. That is, the distribution is not only over the existing classes and there is also a probability that the class of an input is novel without knowing its label. This allows BSAL to consider the change in OSC uncertainty caused by input with uncertain class, whether existing or novel one.

BSAL uses the KL divergence as the loss function. It turns out that by using the KL divergence loss, the mean value of the parameters turns into the Bayesian point estimate [23]. Similar to the Bayesian information theoretic AL proposed in [55], we consider the loss of the predictive posteriors parameterized by $\boldsymbol{\psi}_t$ and $\hat{\boldsymbol{\psi}}_t$:

$$L(\boldsymbol{\psi}_t, \hat{\boldsymbol{\psi}}_t) = \sum_y \int_{\boldsymbol{x}} p(\boldsymbol{x}, y|D_t, \boldsymbol{\psi}_t) \log \frac{p(\boldsymbol{x}, y|D_t, \boldsymbol{\psi}_t)}{p(\boldsymbol{x}, y|D_t, \hat{\boldsymbol{\psi}}_t)} d\boldsymbol{x} \tag{22}$$

$$\hat{\boldsymbol{\psi}}_t = E_{\boldsymbol{\psi}_t \sim p(\boldsymbol{\psi}_t|D_t)}[\boldsymbol{\psi}_t] \tag{23}$$

Our BSAL relies on our proposed Bayesian online semi-supervised classifier (OSC). OSC's parameters, $\boldsymbol{\psi}$, involve the stick-breaking components, the Gaussian components and the hidden configurations. We marginalize out the Gaussian and the stick-breaking components and keep the hidden configurations. The Bayesian point estimate is approximated by the mode of the different configurations induced by the particles. The details on how the discrepancy between the current risk and the current expected risk expressed in Eq. (19) is computed can be found in Appendix F.

As BSAL is an online-based AL, it must assess the data on the-fly and query those which incur highest risk reduction. The problem is how to decide whether the incurred reduction is high or not. A dynamically adaptive threshold, τ, is used so as to request the labels of samples whose current expected risk subtracted from their current risk (see Eq. (19)) breaches the threshold. This latter is adapted using a threshold adjustment step, s, following the Variable Uncertainty strategy in [56]. Further illustration may be found in Alg. 1. The binary input ALE in the algorithm activates/deactivates the active learning.

Another issue is the limited labelling resources. Hence, an optimal querying strategy is needed. To this end, the notion of budget was introduced in [56] in order to estimate the labelling budget. Two counters were maintained: the number of labelled instances $f_t = |X_{L_t}|$ and the budget spent so far: $b_t =$

$\frac{f_t}{|\text{data seen so far}|} = \frac{f_t}{|X_t|}$.

As data arrives, we do not query unless the budget is less than a constant Bd and querying is granted by the sampling model. However, over infinite time horizon this approach will not be effective. The contribution of every query to the budget will diminish over the infinite time and a single labelling action will become less and less sensitive. The authors in [56] propose to compute the budget over fixed memory windows of size *wnd*. To avoid storing the query decisions within the windows, an estimation of f_t and b_t were proposed:

$$\hat{b}_t = \frac{\hat{f}_t}{wnd} \tag{24}$$

where $\hat{f}_t$ is an estimate of how many instances were queried within the last wnd incoming data instances.

$$\hat{f}_t = (1 - 1/wnd)\hat{f}_{t-1} + Lab_{t-1} \tag{25}$$

where $Lab_{t-1} = 1$ if instance x_{t-1} is labelled, and 0 otherwise. Using the forgetting factor $(1-(1/wnd))$, the authors showed that $\hat{b}_t$ is an unbiased estimate of b_t.

In the present paper, this notion of budget is adopted in BSAL so that the labelling rate is controlled. Note that in our experiments, we set $wnd = 100$ as in [56].

Instead of fixing the precision hyper-parameters (also called concentration hyper-parameters) α_1 and α_2, we put hyper priors over them using $G(a,b)$ (gamma priors with shape a and scale b) and sample their values online following the sampling approach in [57]. More details on the sampling routine can be found in Appendix G.

Algorithm 1 Steps of OSC-BSAL

```
**Input:** data stream, OSC hyper-parameters $\{\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, k_0, v_0, a, b\}$ (see Sec. 4.1), memory factor $\lambda$, maximum number of particles $P$, budget $Bd$, $ALE$
**initialize:** set OSC precision parameters $\{\alpha_2, \alpha_{1i}\}_{i=1}^{m_2}$ to 1 (more details in Sec. 4.1 and Appendix G), set the OSC first particle weight $w_0^{(1)}$ to 1, threshold $\tau$ to 0.1, threshold adjustment step $s$ to 0.1, $wnd = 100$, $\hat{f}_t = 0$ (Eq. (24)) and $t = 0$
while (true) do
    $t \leftarrow t + 1$
    extract low level features $\boldsymbol{x_t}$ from the current data samples    ▷ (see Sec. 4)
    if $ALE = 1$ then    ▷ active learning is activated
        if $\hat{b}_t < Bd$ then    ▷ enough budget, Eq. (24)
            $a = \hat{\Delta}(\boldsymbol{x}_t, y_t | D_{t-1}, q_t = 1)$    ▷ (see Eq. (19))
            if $a > \tau$ then
                $Lab_t = 1$
                $y_t \leftarrow query(\boldsymbol{x_t})$
                $\tau = \tau(1 + s)$
            else
                $Lab_t = 0$
                $\tau = \tau(1 - s)$
            end if
        else
            $Lab_t$=0
        end if
    else
        Receive ($Lab_t$)
        if $Lab_t = 1$ then    ▷ instance $\boldsymbol{x_t}$ label is known
            $y_t \leftarrow reveal(\boldsymbol{x_t})$
        end if
    end if
    if $Lab_t = 0$ then
        predict the label of $\boldsymbol{x_t}$    ▷ (see Eq. (1))
        update OSC model with instance $\boldsymbol{x_t}$    ▷ (see Eq. (5), Eq. (8) and Eq. (9))
    else
        update OSC model with instance $\boldsymbol{x_t}$ and label $y_t$    ▷ (see Eq. (3) and Eq. (5))
    end if
    sample new precision parameters $\{\alpha_2, \alpha_{1i}\}_{i=1}^{m_2}$    ▷ (see Alg. 2 in Appendix G)
    compute $\hat{f}_{t+1}$ and update $\hat{b}_{t+1}$    ▷ (see Eq. (25) and Eq. (24))
end while
```

4. Experiments

In this section, we evaluate OSC-BSAL in two steps. In the first step, the active learning (AL) strategy is deactivated ($ALE = 0$) while the classification performance of OSC is evaluated on three datasets: Opportunity, WISDM and SCMA datasets (see below). The data comes as a stream through the feature extractor, then OSC classifies the currently performed activity and updates its particles set. To show the efficiency of the classification besides its fast learning capability, we compare against static offline classification methods: Hoeffding decision tree (DT) and support vector machine (SVM). We also compare against the online method STAR [46] already discussed in Sec. II. In the second step, the active learning strategy, proposed in Sec. 3.2, is activated ($ALE = 1$) to evaluate the whole framework OSC-BSAL on the same datasets. The classification performance is measured according to the average accuracy (AA) which is the correctly classified data samples divided by the total testing data samples. We also compute the average class accuracy (ACA) which is the average of the average accuracies across different activities (classes). This measurement manifests BSAL performance consistency across all activities by implicitly penalising the accuracy when there are misclassifications of infrequent activities. Hence, it illustrates the class discovery performance of BSAL.

The Opportunity (Opp) dataset was the basis of the activity recognition challenge (http://www.opportunity-project.eu/challenge) proposed in the context of the European research project OPPORTUNITY [58]. Opp is a high-dimensional HAR dataset labelled for modes of locomotion, gestures and high-level activities. The data is acquired from four human subjects. In this paper, we use a subset of the dataset corresponding to 3 subjects, denoted by Si and focus on recognition of gesture and modes of locomotion in order to show OSC-BSAL high performance. Details of Opp are presented in Tab.1, where N is the number of instances, d is the number of features/attributes, N_c is the number of classes. Opp was collected from subjects while performing daily activities in a sensor-rich environment of a room akin to an apartment with kitchen, deckchair, and outdoor access. 112 wearable and pervasive sensors with different sensing modalities were used. Each subject executed 4 motion activities and 17 gestures (e.g., *sitting*, *walking*, *standing*, *open fridge*, *clean table*, *move cup*, etc.). The task is to recognise the currently performed activity. Further details can be found in [58]. To extract informative features from this data, we employ a Conditional Restricted Boltzmann Machine (CRBM). Details about CRBM, how it is trained and applied on Opp dataset can be found in Appendix A.

The WISDM dataset is collected from user's mobile phone accelerometer sensor [36]. Six activities are performed by user while data collection, namely, *walking*, *jogging*, *sitting*, *standing*, *upstairs* and *downstairs*. The dataset is collected by different users and contains more than 1 million annotated accelerometer samples. The feature extractor outputs a total of 5424 samples by taking 10 seconds worth of accelerometer samples (200 records/lines in the raw file) and transform them into a single example/tuple of 46 values. Most of these features are simple statistical measures. Details of WISDM are presented in Tab.1. Further details can be found in [36].

Activity Recognition from a Single Chest-Mounted Accelerometer (SCMA) dataset is collected from a wearable accelerometer mounted on the chest from 15 participants [59]. Seven activities are performed by the participants while data collection such as *Standing*, *Walking*, *Working at Computer*, *Talking while Standing*. In this paper, we use a subset of the dataset corresponding to 3 subjects, denoted by *Si*. Details of SCMA are presented in Tab.1. For this dataset, we do not use any feature extraction model. We apply OSC-BSAL directly on the low dimensional features of the data.

All datasets were collected and saved in flat files. To simulate streams from these files, the algorithm reads through the data in the same order it was collected. If BSAL decides to query a certain sample, this latter is sent along with its label to the online classifier in order to update itself. Note that it is assumed that the ground truth is available immediately after a query is made.

The following experiments demonstrate the role of each component of CRBM-OSC-BSAL. We compare SVM and DT trained offline to OSC trained online

Table. 1 Real AR Dataset propreties used for evaluating CRBM-OSC-BSAL

datasets	N	d	N_c
Opp S2	133023	113	4 Locomotions and 17 Gestures
Opp S3	124320	113	4 Locomotions and 17 Gestures
Opp S4	105082	113	4 Locomotions and 17 Gestures
WIDSM	5424	46	6 Locomotions
SCMA S1	162502	3	7 Activities
SCMA S2	138002	3	7 Activities
SCMA S3	102342	3	7 Activities

with no prior knowledge about the data (blind). Results show that OSC outperforms SVM and DT as well as online trained STAR [46] on non-stationary data. We also show that the superiority of OSC over SVM and DT increases with the degree of data non-stationarity. We demonstrate the impact of CRBM by comparing CRBM-OSC to OSC. Results show that CRBM allows better performance, especially on high dimensional data. Comparing CRBM-OSC-BSAL to CRBM-OSC demonstrates the ability of BSAL to improve the performance while using much less data.

4.1. ***Classification performance on Opp data***

In this set of experiments, we evaluate the classification performance of CRBM-OSC on Opp data while active learning (AL) is not considered. The experiments are carried out on three subjects of Opp data with the goal of recognising user's modes of locomotion/gestures. CRBM parameters are set as explained in Appendix A (see Tab. A.15). A hyper-prior can be put over the hyper-parameters of the Normal-Inverse-Wishart prior on cluster parameters (G_0) as in [60]. Alternatively, online non-parametric empirical Bayes can be proposed to find a point estimate of G_0 [61]. However, to keep the computation simple, we chose these hyper-parameters by hand, based on prior knowledge about the scale of the data. The prior mean $\boldsymbol{u}_0$ is set to $\mathbf{0}$. The co-variance matrix $\boldsymbol{\Sigma}_0$ is roughly set to be large relative to the data. We set them to the identity matrix times the distance between the two farthest points in the data. The degree of freedom, v_0, must be greater than the number of dimensions d.

We set it to $d+2$. The hyper-parameter k_0 is empirically set to 0.01. We tested OSC with different parameters $k_0 \in \{1, 0.1, 0.01, 0.05, 0.001\}$ on hold-out Opp data and found that $k_0 = 0.01$ yields to the best performance. The memory factor λ in Eq.(5) is empirically set to 0.95. Similar to k_0, we tested different memory factor (introduced in Eq.(5)) $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1\}$. We noticed that impact of the memory factor is insignificant, even for very high memory factor around 1 i.e., not using memory factor (See. Eq. (5)), OSC still performs very well. This is due to the re-sampling strategy proposed in Sec. 3.1.3. That is, the particles with low probability will be discarded allowing the algorithm to forget obsolete information and adapt to changes. However, setting the memory factor to 0.95 yields the best performance.

4.1.1. ***Locomotion***

In order to show the efficiency of the proposed online algorithm, we compare it to well known offline classification models. DT (hoeffding decision tree) and SVM (support vector machines with polynomial kernel, degree 3) have been efficiently applied for HAR in static environments [11]. We run two experiments with DT and SVM using two different training settings. In *setting 1*, SVM and DT are built from the datasets of all subjects excluding the one whose data is used for the evaluation. The results are presented in Tab. 2. In *setting 2*, training and testing are done using the data of the same subject with two-fold cross-validation; we split the data into two sets training and testing. The samples in each set are selected using indices chosen randomly over the whole data indices. We train and test then alternate the training and testing sets. This step is repeated 15 times where the results for each subject are averaged over 15*2 runs (see Tab. 3). We run SVM and DT with Weka 3.8 [62] (SVM and DT use Weka default parameters). Although, DT and SVM are trained with more data in *setting 1*, all the results shown in *setting 2* are better. This variance may be explained by the distinct motion styles of the subjects. We adopt *setting 2* in the upcoming comparison. Next, we train our model CRBM-OSC online using *setting 2* (see Tab. 3). It can be seen that CRBM-OSC performs better

Table. 2 Classification performance for locomotion activities under *setting 1*

Subject	Method	AA	Stand	Walk	Lie	Sit	ACA
S2	DT(%)	64.3	40	81.4	0	83.8	51.3
	SVM(%)	64.7	29.7	97.8	1.2	84	53.2
S3	DT(%)	41.9	27	95.6	0	0	41.1
	SVM(%)	73.6	84	59.7	0	68.4	53
S4	DT(%)	56.6	37.8	89.8	0	83.2	52.7
	SVM(%)	72.5	92	10.8	96.5	96.3	73.9

Table. 3 Classification performance for locomotion activities under *setting 2*

Subject	Method	AA	Stand	Walk	Lie	Sit	ACA
S2	CRBM-OSC(%)	**96**	**95.7**	**91.4**	98.4	**99.8**	**96.3**
	DT(%)	89.5	89.8	78.9	89.7	96.5	88.7
	SVM(%)	91.2	90.1	79.6	**99.3**	99.6	92.2
S3	CRBM-OSC(%)	**95.2**	**96.2**	**92.1**	98.3	**96.5**	**95.8**
	DT(%)	83.6	86.1	74.8	97.9	89.3	87
	SVM(%)	85	93.7	60.4	**99.9**	96.4	88.1
S4	CRBM-OSC(%)	**94.2**	**94.1**	**90.2**	99.3	**98.5**	**95.5**
	DT(%)	87.2	87.8	81.1	96	91.9	89.2
	SVM(%)	88	93.4	67.1	**99.9**	98.3	89.7
All	CRBM-OSC(%)	**96**	**95.7**	**94**	98.7	**98.4**	**96.7**
	DT(%)	83.3	94	56.6	96.9	88.6	84
	SVM(%)	84.5	92	60.7	**99.1**	96.6	87.2

on all subsets of the data and for almost all activities. A reason for CRBM-OSC superiority may be explained by the ability of the algorithm to cope with dynamic changes in the individual activities. Based on this analysis and the previous one regarding the difference motion styles among the subjects, we can expect that the performance superiority of CRBM-OSC will increase if training is done on the datasets of all subjects. Thus, we run an experiment on the datasets of all subjects together using *setting 2* (see Tab. 3). It can be clearly seen that CRBM-OSC outperforms SVM and DT by far when all-subject data is considered compared to the case where only one-subject data is used.

Note that the performance of CRBM-OSC is consistent across all activities as its average class accuracy (ACA) is high. However, ACA is slightly better than the average accuracy (AA) for all methods. This is due to the plenty of available labels. In fact, the challenge lies in maintaining high ACA with few

Table. 4 Classification performance for locomotion activities

Subject	Method	AA	Stand	Walk	Lie	Sit	ACA
S2	CRBM-OSC(%)	**96**	95.7	**91.4**	**98.4**	**99.8**	**96.3**
	OSC(%)	93.52	**98.7**	77.8	99	89.7	91.3
S3	CRBM-OSC(%)	**95.2**	96.2	**92.1**	**98.3**	96.5	**95.8**
	OSC(%)	94.3	**99.1**	82.2	98.1	**97.7**	94.3
S4	CRBM-OSC(%)	**94.2**	94.1	**90.2**	**99.3**	**98.5**	**95.5**
	OSC(%)	92.6	**98.9**	74.9	96.5	98.2	92.1
All	CRBM-OSC(%)	**96**	**95.7**	94	**98.7**	**98.4**	**96.7**
	OSC(%)	93.8	80.3	**98.2**	88.6	97.1	91

labels. Such challenge can be met by BSAL (to be discussed in the next section).

In order to show the effect of the feature extractor, we train OSC online using *setting 2*, where CRBM is not considered (see Tab. 4). Generally, CRBM-OSC performs better in terms of average accuracy and average class accuracy. In addition, CRBM reduces the feature dimension from 113 to 10 which requires less computation and memory resources to process the streaming samples. We believe that with less discriminative low level features and more complex activities, CRBM will play more prominent role.

4.1.2. ***Gestures***

Like the previous section (Sec. 4.1.1), we compare the performance of DT and SVM under the two training settings. SVM and DT are run with Weka 3.8 [62] and the results for each subject are averaged over 15*2 runs. The results obtained under both settings are presented in Tab. 5 and Tab. 6. Like in Sec. 4.1.1, DT and SVM, trained with less data under *setting 2*, show better performance than that of DT and SVM trained under *setting 1.* This variance confirms the assumption drawn in Sec. 4.1.1 that there are changes in the data across different subjects. We will adopt *setting 2* in the upcoming comparisons.

Next, we train our model CRBM-OSC online with two-fold cross-validation. As in Sec. 4.1.1, the results for each subject are averaged over 15*2 runs (see Tab. 6). CRBM-OSC average accuracy is better than SVM and DT's one on all the datasets. In addition, CRBM-OSC average accuracies are the best for the majority of the activities. Interestingly, the performance of CRBM-OSC

Table. 5 Classification performance for gesture activities under *setting 1*

Subject	Method	AA	Open Door1	Open Door2	Close Door1	Close Door2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer1	Close Drawer1	Open Drawer2	Close Drawer2	Open Drawer3	Close Drawer3	Clean Table	Drink Cup	Toggle Switch	ACA
S2	DT(%)	34	3.3	17.9	0	3.6	6.3	0	3.4	0.6	13.3	0	0	27.5	75.1	0	27.8	64.3	0	18.1
	SVM(%)	40	7.2	1	5.7	59.3	11.2	5.3	24	60.5	9.8	0	25.8	69.5	72.8	7.6	24.6	61.1	5.5	26.5
S3	DT(%)	51.3	11.1	2.2	8.7	67.4	6.8	43.3	45.9	10.7	57.6	65	62.2	17	77.8	0.3	59.7	78.7	92.3	37.8
	SVM(%)	40	18.5	7	0	66.1	1.1	1.1	28.3	0.7	12	16.7	0.2	3	92.9	0	64.2	67.1	93.8	27.8
S4	DT(%)	36.1	0	0	0	0	1.4	0	1.2	39.4	39.5	0.1	0.5	0	0	0	59.4	88.7	59.5	17
	SVM(%)	45	1.6	60.6	1.5	56.5	40.9	33.4	24.9	42.2	0	36.3	36.7	59.9	62.2	46.5	6	63.6	94.6	39.3

Table. 6 Classification performance for gesture activities under *setting 2*

Subject	Method	AA	Open Door1	Open Door2	Close Door1	Close Door2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer1	Close Drawer1	Open Drawer2	Close Drawer2	Open Drawer3	Close Drawer3	Clean Table	Drink Cup	Toggle Switch	ACA
S2	CRBM-OSC(%)	**95**	**93.6**	**96**	**94.7**	**94.8**	93.6	**96.1**	**93.6**	90.1	**92.1**	**94.9**	**93.6**	**91.6**	93.4	91.3	95	97.1	93.4	**93.8**
	DT(%)	72.2	47.3	80.1	58.3	76.1	85.2	63.5	71.8	89.9	41.6	52.8	42.4	70.6	64.1	52	56.1	84	91.2	66.3
	SVM(%)	**95**	91.8	82.3	83	94.4	**95.9**	74	90.5	**99.2**	90.9	94.7	81.4	82.7	**98.4**	**93**	**97.6**	**99**	**97**	90.9
S3	CRBM-OSC(%)	**94.4**	**93.3**	**95.4**	**92.3**	**92.3**	**94.6**	**94.2**	**93.9**	94.2	**94**	**95**	**93.2**	**92.5**	**93.9**	**95.2**	**92.5**	96.2	92.8	**93.9**
	DT(%)	66.6	33.5	19.3	73.3	89.1	86.3	35.2	77.8	84.7	58.3	81	4.5	37.1	62.3	73.9	60.5	79	97.6	65.5
	SVM(%)	90.3	90.9	77.6	80.9	88.9	88.9	71.4	88.6	**98.2**	85.5	83.6	77.1	87.3	91.8	78.2	90	**98.7**	**99**	86.9
S4	CRBM-OSC(%)	**94.5**	**93.6**	**93.9**	**94**	**93.3**	**93.7**	**94.2**	**92.5**	92.3	**93.2**	**95.5**	**93.9**	**93.4**	**94.8**	**92.7**	**93.4**	96.7	93.2	**93.8**
	DT(%)	68.7	53.5	43.9	41.3	87.6	91.1	41.6	51	86.7	56.9	85.5	83.4	49	42.2	42.2	59.1	83.7	97.1	64.5
	SVM(%)	91.8	93.2	91.4	82.6	85.4	91.1	83.1	79.7	**98.5**	88.9	89.6	86.4	87.6	90.2	92.5	84.6	**99.5**	**99.5**	89.6
All	CRBM-OSC(%)	**94.9**	**92.5**	**93.9**	**94.6**	**94.1**	**94.6**	**94.7**	**93.3**	93.5	**93.2**	**94.7**	**94.7**	**93.7**	**94.8**	**95.2**	**93.6**	93.9	**97.2**	**94.3**
	DT(%)	57.1	24.2	18.9	61.6	81.8	86.6	46.4	46.4	43	21.6	84.7	33.1	42.2	30	38.4	47.8	75.9	85.7	51.1
	SVM(%)	84.5	79.3	74.1	64.9	82.2	82.9	66.2	75.8	**94.3**	70.3	76.3	73.2	80.4	79.7	75.7	82.9	**97**	92.1	79.3

increases when training is done on the datasets of all subjects. Hence, the analysis regarding the importance of adaptive learning is demonstrated for gesture activities too.

Unlike the case with the locomotion activities, we can notice that ACA is slightly lower than the average accuracy (AA) for all methods. This can be explained by the increase in the number of activities meaning that the same number of labels is distributed on larger number of classes. Thus, the risk of misclassifying the least frequent activities may increase. Noticeably, ACA values for SVM and DT decrease more significantly than that for CRBM-OSC. Such behaviour may be explained by the fact that OSC is a semi-supervised learning algorithm. That is, it uses both unlabelled and labelled data which curbs the consequence of scarce labels for certain activities.

4.2. ***Classification performance on WISDM data***

In this set of experiments, we evaluate the classification performance of OSC on WISDM data while active learning (AL) is not considered. OSCs' parameters are set the same way as in the previous experiments (see Sec. 4.1).

Table. 7 Classification performance on WISDM under *setting 1*

Method	AA	Walking	Jogging	Upstairs	Downstairs	Sitting	Standing	ACA
DT(%)	70.7	78.4	92.6	22.1	22.3	92.3	64.7	62
SVM(%)	69	80.9	91.8	13.1	19	76.2	66	69.3

Table. 8 Classification performance on WISDM under *setting 2*

Method	AA	Walking	Jogging	Upstairs	Downstairs	Sitting	Standing	ACA
OSC (%)	**88.1**	95.2	93.4	**69.3**	**64.9**	86.5	75.9	**80.9**
DT(%)	74,8	88	93.1	17.7	22.2	93.8	79.3	65.68
SVM(%)	84.4	**96.1**	**99.1**	47	28.4	**96.1**	**91.5**	76.36

Because WISDM data is collected from 36 different users, activities are performed with different styles. To show the personalisation impact on the data, we, similar to the previous section, run two experiments with DT and SVM using two different training settings. In *setting 1*, SVM and DT are built from the first 50% samples which involves different subjects from the last 50% samples used for the evaluation. The results are presented in Tab. 7. In *setting 2*, training and testing are done with two-fold cross-validation. As in Sec. 4.1.1, the results are averaged over 15*2 runs (see Tab. 8).

All the results of SVM and DT shown in *setting 2* are better than those shown in *setting 1.* This variance demonstrates that there are changes in the date caused by involving different users over time. We adopt *setting 2* in the upcoming comparison.

Next, we train our model OSC online using *setting 2* (see Tab. 8). It can be seen that OSC average accuracy is the best. As on Opp data, the superiority of OSC can be explained by the ability of the algorithm to cope with dynamic changes. These changes occur in the activities of the same subject and mainly across different subjects. Note that the performance of OSC is consistent across all activities as its average class accuracy (ACA) is the highest.

4.3. Classification performance on SCMA data

In this set of experiments, we evaluate the classification performance of OSC on SCMA data while active learning (AL) is not considered. OSCs' parameters are set the same way as in the previous experiments (see Sec. 4.1).

Table. 9 Classification performance on SCMA under *setting 2*

Subject	Method	AA	Working at Computer	Standing Up, Walking and Going up\down stairs	Standing	Walking	Going Up \Down Stairs	Walking and Talking with Someone	Talking while Standing	ACA
	OSC (%)	**99.99**	99.99	**99.79**	**99.95**	**99.99**	**99.84**	**99.89**	**99.99**	**99.92**
S1	DT(%)	98.8	99.99	99.1	92.9	97.2	91.7	99.1	99.99	97.1
	SVM(%)	93.87	99.7	10.1	60.7	94.3	75.3	30.2	99.89	67.1
	OSC (%)	**99**	99.99	**99.94**	**99.96**	**99.99**	99.89	99.98	99.99	**99.96**
S2	DT(%)	99.6	99.99	98	98.2	99.9	**99.99**	99.7	99.99	99.39
	SVM(%)	83.41	99.99	81.5	46.5	75.2	9.4	**99.99**	99.99	73.2
	OSC (%)	**99.98**	99.99	**99.98**	**99.79**	99.98	**99.92**	**99.92**	99.99	**99.93**
S3	DT(%)	99.79	99.99	97.9	99	99.99	99.5	98.9	99.99	99.32
	SVM(%)	93.55	99.99	89.2	19.3	99.99	62.3	12.9	99.3	69
	OSC (%)	**99.98**	**99.99**	**99.92**	**99.94**	**99.98**	**99.92**	**99.93**	**99.98**	**99.95**
All	DT(%)	75.69	96.3	12.8	42.5	75.7	15.1	12.8	83.3	59.17
	SVM(%)	51.98	59.2	17.2	14.3	19.3	13.7	16.9	98.6	34.2

As in the case of Opp data, SCMA data is acquired from different subjects. Hence, activities across the subjects are performed with different styles. Relying on the comparisons done for Opp data and the similarity of the data collection between Opp and SCMA, we skip the comparison of OSC performance between *setting 1* and *setting 2* and adopt *setting 2* in the upcoming comparison. Therefore, OSC is trained online using *setting 2* (see Tab. 9).

The results in Tab. 9 show that OSC maintains the good performance shown on Opp and WISDM datasets. As on Opp data, the superiority of OSC can be explained by the ability of the algorithm to cope with dynamic changes in the activities. Note that the performance of OSC is consistent across all activities as its average class accuracy (ACA) is the highest. Similar to Opp data, the activity styles vary among the subjects. Thus, we can expect that the performance superiority of OSC will increase if training is done on the datasets of all subjects. Thus, we run an experiment on the datasets of all subjects together using *setting 2* (see Tab. 9). It can be clearly seen that OSC outperforms SVM and DT by far when all-subject data is considered compared to the case where only one-subject data is used.

4.4. ***Active learning performance on Opp data***

In this section, we evaluate the performance of the whole model including the active learning strategy on Opp data. The evaluation of CRBM-OSC-BSAL

Table. 10 Classification performance for locomotion activities (5%)

Subject	Method	AA	Stand	Walk	Lie	Sit	ACA
S2	CRBM-OSC-BSAL(%)	93	88	90.1	93.2	99.3	92.7
	CRBM-OSC(%)	**96**	95.7	91.4	98.4	99.8	**96.3**
	STAR(%)	74.9	82.5	50.1	89.9	47.3	67.5
	DT(%)	89.5	89.8	78.9	89.7	96.5	88.7
	SVM(%)	91.2	90.1	79.6	99.3	99.6	92.2
S3	CRBM-OSC-BSAL(%)	86	88	81.5	98.3	84.2	88
	CRBM-OSC(%)	**95.2**	96.2	92.1	98.3	96.5	**95.8**
	STAR(%)	68.6	87	44.3	86.1	7.2	56.2
	DT(%)	83.6	86.1	74.8	97.9	89.3	87
	SVM(%)	85	93.7	60.4	99.9	96.4	88.1
S4	CRBM-OSC-BSAL(%)	91.4	91.8	85.4	92.9	97.9	92
	CRBM-OSC(%)	**94.2**	94.1	90.2	99.3	98.5	**95.5**
	STAR(%)	71.1	84.2	43.6	90	30.3	62
	DT(%)	87.2	87.8	81.1	96	91.9	89.2
	SVM(%)	88	93.4	67.1	99.9	98.3	89.7
All	CRBM-OSC-BSAL(%)	93.6	93.6	90.8	96.5	97.6	94.6
	CRBM-OSC(%)	**96**	95.7	94	98.7	98.4	**96.7**
	DT(%)	83.3	94	56.6	96.9	88.6	84
	SVM(%)	84.5	92	60.7	99.1	96.6	87.2

is based on a prequential methodology: each time we get an instance, first we test it, and if we decide to incur the cost of its label, then we use it to train the classifier [56]. The results are average over 30 runs. The parameters are set as in Sec. 4.1. Results obtained in the previous section (Tab. 3 and Tab. 6) are used for performance comparison.

4.4.1. ***Locomotion***

We compare the results obtained in Sec. 4.1.1 (Tab. 3) to the ones of CRBM-OSC-BSAL with few queried data samples, around 5% for each subject. We also compare against the online method STAR [46] (already discussed in Sec. 2) which also queries 5% of the processed data. The results are shown in Tab. 10.

CRBM-OSC-BSAL shows better average accuracy (AA) than all competitors excluding CRBM-OSC. However, the AL strategy BSAL has reduced the number of labelled samples from 50% to around 5% while leading to an average (over all datasets) of arround 4.3% less accuracy compared to CRBM-OSC.

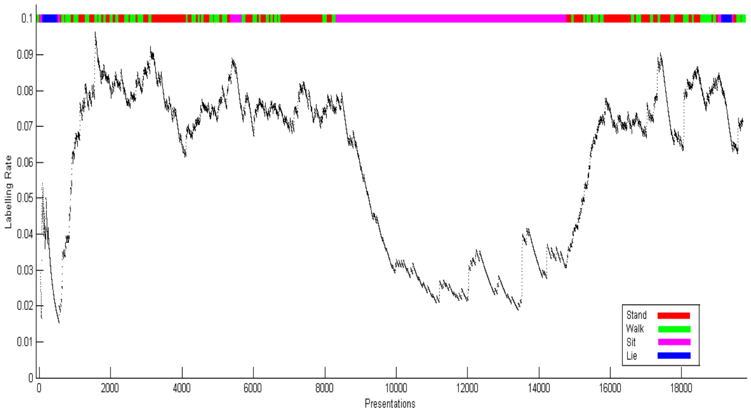


Figure. 3 Labelling rate along the stream of subject 2 (S2)

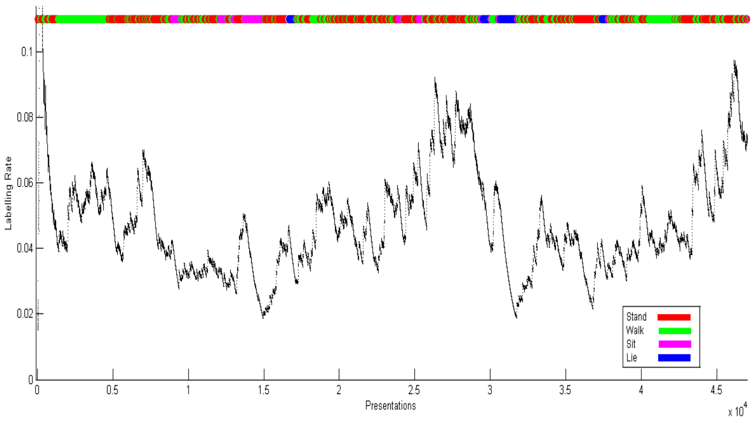


Figure. 4 Labelling rate along the stream of subject 3 (S3)

CRBM-OSC-BSAL significantly outperforms STAR which also employs an AL strategy to query 5% of the data. Moreover, CRBM-OSC-BSAL outperforms SVM and DT even though the percentage of labels used for training is 45%

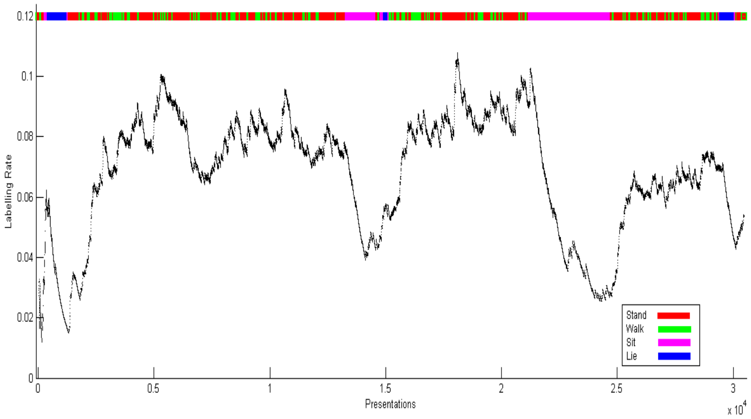


Figure. 5 Labelling rate along the stream of subject 4 (S4)

higher. Similar to the results presented in Sec. 4.1, the performance of CRBM-OSC-BSAL is more significant when training is done on the datasets of all subjects. Hence, the proposed AL is capable of maintaining high performance when data evolves more substantially. Note that BSAL is able to maintain consistent performance across all activities. Indeed, the average class accuracy (ACA) is still high even though number of labels is 45% less than the one for CRBM-OSC. However, we can notice that ACA is not as better than AA as it is for CRBM-OSC. Though, this is normal as the number of labels has dramatically decreased. Therefore, it is a strong point of BSAL to keep the ACA high (even higher than AA) with few labels.

In order to visualize the behaviour of BSAL, we draw the prequential labelling rate as the data streams pass though CRBM-OSC-BSAL (see Fig. 3, Fig. 4 and Fig. 5). To smooth the curve, a fading factor of 0.999 is used to compute the learning rate. Some observations on BSAL behaviour can be deduced from Fig. 3, Fig. 4 and Fig. 5. First, BSAL tends to query the activities appearing for the first time, then the labelling rate falls down. For instance, the labelling rate across *lying* and *sitting* activities is high when the activities first appear. In the second appearance of the same activities, the labelling rate is not

Table. 11 Classification performance for gesture activities (5%)

Subject	Method	AA	Open Door1	Open Door2	Close Door1	Close Door2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer1	Close Drawer1	Open Drawer2	Close Drawer2	Open Drawer3	Close Drawer3	Clean Table	Drink Cup	Toggle Switch	ACA
S2	CRBM-OSC-BSAL(%)	85.2	81.3	55.8	77.1	96	96.8	76.5	89.7	90.9	95.7	86.7	95	96.8	96.6	95.4	94.4	77.9	97.9	82.6
	CRBM-OSC(%)	**95**	93.6	96	94.7	94.8	93.6	96.1	93.6	90.1	92.1	94.9	93.6	91.6	93.4	91.3	95	97.1	93.4	**93.8**
	DT(%)	72.2	47.3	80.1	58.3	76.1	85.2	63.5	71.8	89.9	41.6	52.8	42.4	70.6	64.1	52	56.1	84	91.2	66.3
	SVM(%)	**95**	91.8	82.3	83	94.4	95.9	74	90.5	99.2	90.9	94.7	81.4	82.7	98.4	93	97.6	99	97	90.9
S3	CRBM-OSC-BSAL(%)	91.1	87.6	84.2	72.7	95.8	94.1	80.1	88.6	92.3	93.7	88.7	84.5	93.9	95.4	88.2	95.1	93	94.7	89.5
	CRBM-OSC(%)	**94.4**	93.3	95.4	92.3	92.3	94.6	94.2	93.9	94.2	94	95	93.2	92.5	93.9	95.2	92.5	96.2	92.8	**93.9**
	DT(%)	66.6	33.5	19.3	73.3	89.1	86.3	35.2	77.8	84.7	58.3	81	4.5	37.1	62.3	73.9	60.5	79	97.6	65.6
	SVM(%)	90.3	90.9	77.6	80.9	88.9	88.9	71.4	88.6	98.2	85.5	83.6	77.1	87.3	91.8	78.2	90	98.7	99	86.9
S4	CRBM-OSC-BSAL(%)	85.6	77.7	53.9	57.5	91.7	92	53.5	85.1	87.5	84.3	73	62.7	88.9	83	84.8	91.9	97	90.5	79.7
	CRBM-OSC(%)	**94.5**	93.6	93.9	94	93.3	93.7	94.2	92.5	92.3	93.2	95.5	93.9	93.4	94.8	92.7	93.4	96.7	93.2	**93.8**
	DT(%)	68.7	53.5	43.9	41.3	87.6	91.1	41.6	51	86.7	56.9	85.5	83.4	49	42.2	42.2	59.1	83.7	97.1	64.5
	SVM(%)	91.8	93.2	91.4	82.6	85.4	91.1	83.1	79.7	98.5	88.9	89.6	86.4	87.6	90.2	92.5	84.6	99.5	99.5	89.6
All	CRBM-OSC-BSAL(%)	89.6	82.4	82.4	66	64.8	95.4	93.4	73.1	86.8	87.9	91	84	80.5	90	92.3	87	93.9	98	80.9
	CRBM-OSC(%)	**94.9**	92.5	93.9	94.6	94.1	94.6	94.7	93.3	93.5	93.2	94.7	94.7	93.7	94.8	95.2	93.6	93.9	97.2	**94.3**
	DT(%)	57.1	24.2	18.9	61.6	81.8	86.6	46.4	46.4	43	21.6	84.7	33.1	42.2	30	38.4	47.8	75.9	85.7	51.1
	SVM(%)	84.5	79.3	74.1	64.9	82.2	82.9	66.2	75.8	94.3	70.3	76.3	73.2	80.4	79.7	75.7	82.9	97	92.1	79.3

Table. 12 Classification performance for gesture activities (10%)

Subject	Method	AA	Open Door1	Open Door2	Close Door1	Close Door2	Open Fridge	Close Fridge	Open Dishwasher	Close Dishwasher	Open Drawer1	Close Drawer1	Open Drawer2	Close Drawer2	Open Drawer3	Close Drawer3	Clean Table	Drink Cup	Toggle Switch	ACA
S2	CRBM-OSC-BSAL(%)	**97.2**	86.5	81.1	87.8	98.8	98.1	91.9	95.3	96.2	97.8	97.9	98.2	98.7	98.5	95.4	97.7	99.6	100	**95.3**
	CRBM-OSC(%)	95	93.6	96	94.7	94.8	93.6	96.1	93.6	90.1	92.1	94.9	93.6	91.6	93.4	91.3	95	97.1	93.4	93.8
	DT(%)	72.2	47.3	80.1	58.3	76.1	85.2	63.5	71.8	89.9	41.6	52.8	42.4	70.6	64.1	52	56.1	84	91.2	66.3
	SVM(%)	95	91.8	82.3	83	94.4	95.9	74	90.5	99.2	90.9	94.7	81.4	82.7	98.4	93	97.6	99	97	90.9
S3	CRBM-OSC-BSAL(%)	94	90.2	86.8	75.6	95.7	95.5	82.4	91	94	94.5	92.1	83.4	95.1	96.5	91	95.4	98.8	94.9	91.4
	CRBM-OSC(%)	**94.4**	93.3	95.4	92.3	92.3	94.6	94.2	93.9	94.2	94	95	93.2	92.5	93.9	95.2	92.5	96.2	92.8	**93.9**
	DT(%)	66.6	33.5	19.3	73.3	89.1	86.3	35.2	77.8	84.7	58.3	81	4.5	37.1	62.3	73.9	60.5	79	97.6	65.6
	SVM(%)	90.3	90.9	77.6	80.9	88.9	88.9	71.4	88.6	98.2	85.5	83.6	77.1	87.3	91.8	78.2	90	98.7	99	86.9
S4	CRBM-OSC-BSAL(%)	94.2	88.2	86.9	82.3	96.3	95.1	84.8	93.9	93.9	93	89	85.7	96.1	94.5	96.2	94.8	98.9	94.4	92
	CRBM-OSC(%)	**94.5**	93.6	93.9	94	93.3	93.7	94.2	92.5	92.3	93.2	95.5	93.9	93.4	94.8	92.7	93.4	96.7	93.2	**93.8**
	DT(%)	68.7	53.5	43.9	41.3	87.6	91.1	41.6	51	86.7	56.9	85.5	83.4	49	42.2	42.2	59.1	83.7	97.1	64.5
	SVM(%)	91.8	93.2	91.4	82.6	85.4	91.1	83.1	79.7	98.5	88.9	89.6	86.4	87.6	90.2	92.5	84.6	99.5	99.5	89.6
All	CRBM-OSC-BSAL(%)	91.4	84.6	87.6	69	74	96	95.4	76.7	90.5	92.4	93.4	83.2	83.9	91.8	94.3	87.7	94.6	98.4	88
	CRBM-OSC(%)	**94.9**	92.5	93.9	94.6	94.1	94.6	94.7	93.3	93.5	93.2	94.7	94.7	93.7	94.8	95.2	93.6	93.9	97.2	**94.3**
	DT(%)	57.1	24.2	18.9	61.6	81.8	86.6	46.4	46.4	43	21.6	84.7	33.1	42.2	30	38.4	47.8	75.9	85.7	51.1
	SVM(%)	84.5	79.3	74.1	64.9	82.2	82.9	66.2	75.8	94.3	70.3	76.3	73.2	80.4	79.7	75.7	82.9	97	92.1	79.3

triggered. Second, long lasting activities are infrequently queried which maintains the performance over evolving streams. Third, we notice that sometimes the labelling rate across *standing* activity suddenly grows. Such behaviour may be caused by the gesture activities which are usually performed while the user is standing. Hence, BSAL expects change or emergence in the activities leading to an increase in the labelling rate. *Walking* activity is peculiar because it is the most frequent and prone to change more than the others. However, when it lasts for considerable duration, labelling rate decreases.

4.4.2. ***Gestures***

We compare the results obtained in Sec. 4.1.2 to the ones of CRBM-OSC-BSAL with few queried data samples, around 5% for each subject (see Tab. 11).

Table. 13 Classification performance on WISDM (10%)

Method	AA	Walking	Jogging	Upstairs	Downstairs	Sitting	Standing	ACA
OSC-BSAL (%)	87.8	92.3	92.9	72.8	62.2	86.2	81.6	**81.3**
OSC (%)	**88.1**	95.2	93.4	69.3	64.9	86.5	75.9	80.9
DT(%)	74,8	88	93.1	17.7	22.2	93.8	79.3	65.68
SVM(%)	84.4	96.1	99.1	47	28.4	96.1	91.5	76.36

Although, CRBM-OSC-BSAL uses only 5% labels for a relatively high number of activities, its AA for S3 is better than the one of SVM and DT which use 50% labels. SVM outperforms CRBM-OSC-BSAL for S2 and S4, but at the cost of high labelling (45% more labels). Nevertheless, CRBM-OSC-BSAL shows better AA results when the datasets of all subjects are used. This highlights BSAL's ability to maintain high performance under more aggressive changes.

Predictably, CRBM-OSC-BSAL's ACA drops in comparison with CRBM-OSC's ACA as the number of labels has dramatically decreased. In addition, the relatively large number of activities makes it harder to maintain consistent high accuracy across all activities. Thus, we run another experiment with 10% of the data samples queried (see Tab. 12). It can be seen that CRBM-OSC-BSAL's ACA surpasses the one of SVM and DT that are trained with 40% more data. Furthermore, CRBM-OSC-BSAL's ACA becomes comparable to CRBM-OSC's. We can also notice that AA has improved and CRBM-OSC-BSAL outperforms SVM and DT for all subjects and has almost the same AA as CRBM-OSC trained with 40% more data.

*4.5. **Active learning performance on WISDM data***

In this section, we evaluate the performance of OSC-BSAL On WISDM data. We take the same evaluation setting as in the previous section (see Se. 4.4). We compare the results obtained in Sec. 4.2 (Tab. 8) to the ones of OSC-BSAL with few queried data samples, around 10%. The results are shown in Tab. 13.

OSC-BSAL shows better average accuracy (AA) than all competitors excluding OSC. However, the AL strategy BSAL has reduced the number of labelled samples from 50% to around 10% while leading to an average (over all datasets) of arround 0.3% less accuracy compared to OSC. OSC-BSAL outperforms SVM

Table. 14 Classification performance on SCMA (10%)

Subject	Method	AA	Working at Computer	Standing Up, Walking and Going up\down stairs	Standing	Walking	Going Up \Down Stairs	Walking and Talking with Someone	Talking while Standing	ACA
S1	OSC-BSAL (%)	99.92	99.93	99.65	99.24	99.99	99.55	99.89	99.99	99.75
	OSC (%)	**99.99**	99.99	99.79	99.95	99.99	99.84	99.89	99.99	**99.92**
	DT(%)	98.8	99.99	99.1	92.9	97.2	91.7	99.1	99.99	97.1
	SVM(%)	93.87	99.7	10.1	60.7	94.3	75.3	30.2	99.89	67.1
S2	OSC-BSAL (%)	**99.94**	99.99	99.87	99.71	99.99	99.99	99.97	99.99	99.93
	OSC (%)	99	99.99	99.94	99.96	99.99	99.89	99.98	99.99	**99.96**
	DT(%)	99.6	99.99	98	98.2	99.9	99.99	99.7	99.99	99.39
	SVM(%)	83.41	99.99	81.5	46.5	75.2	9.4	99.99	99.99	73.2
S3	OSC-BSAL (%)	99.6	99.97	99.99	91.47	99.99	99.25	99.76	99.97	98.62
	OSC (%)	**99.98**	99.99	99.98	99.79	99.98	99.92	99.92	99.99	**99.93**
	DT(%)	99.79	99.99	97.9	99	99.99	99.5	98.9	99.99	99.32
	SVM(%)	93.55	99.99	89.2	19.3	99.99	62.3	12.9	99.3	69
All	OSC-BSAL (%)	99.88	99.97	99.89	99.14	99.99	99.7	99.8	99.6	99.73
	OSC (%)	**99.98**	99.99	99.92	99.94	99.98	99.92	99.93	99.98	**99.95**
	DT(%)	75.69	96.3	12.8	42.5	75.7	15.1	12.8	83.3	59.17
	SVM(%)	51.98	59.2	17.2	14.3	19.3	13.7	16.9	98.6	34.2

and DT even though the percentage of labels used for training is 40% higher. Note that BSAL is able to maintain consistent performance across all activities. Indeed, the average class accuracy (ACA) is the highest even though number of labels is 40% less than the one for OSC, DT and SVM.

4.6. Active learning performance on SCMA data

In this section, we evaluate the performance of OSC-BSAL On SCMA data. We take the same evaluation setting as in the previous section (see Se. 4.4). We compare the results obtained in Sec. 4.3 (Tab. 9) to the ones of OSC-BSAL with few queried data samples, around 10%. The results are shown in Tab. 14. Similar to previous experiments, OSC-BSAL shows good performance compared to the competitors knowing that only 10% of the data is labeled. Similar to the results presented in Sec. 4.1, the performance of OSC-BSAL is more significant when training is done on the datasets of all subjects. Hence, the proposed AL is capable of maintaining high performance when data evolves more substantially.

5. Conclusion and future work

In this paper, we proposed a new learning model composed of a feature extractor (CRBM), an online semi-supervised classifier (OSC) and an active learning algorithm (BSAL) to cope the challenges of human activity recognition from data streams in the smart-home setting. CRBM helps overcome the weary features hand-crafting by learning generic features from unlabelled high-dimensional sensory input. OSC online learns the human activities from stream of generic features. BSAL queries the activities that are expected to bring crucial information for OSC. Experimental results on real-world activity recognition datasets showed the effectiveness of the proposed model. It is worthwhile to point out that the proposed model can be used for different applications where there exists sequential dependency in the data.

In the future, we foresee four directions for research to improve the obtained results and provide more features: (i) different features extraction can be investigated such as Conditional Random Fields (CRFs). We will also seek to unify the training of the feature extraction with OSC-BSAL and accommodate the feature extractor pre-training step into the online setting. (ii) Improve the online learning model to consider co-occurring activities, infer the activity length and performs segmentation. (iii) Investigating Bayesian approach to unify the budget control the Bayesian AL. (iiii) Exploit the proposed model in new applications such as industrial maintenance.

Acknowledgment

A. Bouchachia was supported by the European Commission under the Horizon 2020 Grant 687691 related to the project: *PROTEUS: Scalable Online Machine Learning for Predictive Analytics and Real-Time Interactive Visualization.*

References

[1] B. Longstaff, S. Reddy, D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised learning, in: 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, IEEE, 2010, pp. 1–7.

[2] J. Yang, Toward physical activity diary: motion recognition using simple acceleration features with mobile phones, in: Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics, ACM, 2009, pp. 1–10.

[3] G. Singla, D. J. Cook, M. Schmitter-Edgecombe, Recognizing independent and joint activities among multiple residents in smart environments, Journal of ambient intelligence and humanized computing 1 (1) (2010) 57–63.

[4] D. J. Cook, M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods of information in medicine 48 (5) (2009) 480.

[5] A. Bouchachia, C. Vanaret, GT2FC: An online growing interval type-2 self-learning fuzzy classifier, IEEE Transactions on Fuzzy Systems 22 (4) (2014) 999–1018.

[6] A. Bouchachia, An evolving classification cascade with self-learning, Evolving Systems 1 (3) (2010) 143–160.

[7] A. Bouchachia, Fuzzy classification in dynamic environments, Soft Computing 15 (5) (2011) 1009–1022.

[8] R. Poppe, A survey on vision-based human action recognition, Image and vision computing 28 (6) (2010) 976–990.

[9] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Computer vision and image understanding 104 (2) (2006) 90–126.

[10] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer vision and image understanding 115 (2) (2011) 224–241.

[11] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Computing Surveys (CSUR) 46 (3) (2014) 33.

[12] N. Twomey, T. Diethe, X. Fafoutis, A. Elsts, R. McConville, P. Flach, I. Craddock, A comprehensive study of activity recognition using accelerometers, in: Informatics, Vol. 5, Multidisciplinary Digital Publishing Institute, 2018, p. 27.

[13] S. Wang, G. Zhou, A review on radio based activity recognition, Digital Communications and Networks 1 (1) (2015) 20–29.

[14] B. Settles, Active learning literature survey, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009).

[15] E. Lughofer, On-line active learning: a new paradigm to improve practical useability of data stream modeling methods, Information Sciences 415 (2017) 356–376.

[16] D. D. Lewis, W. A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag New York, Inc., 1994, pp. 3–12.

[17] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 287–294.

[18] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, 2008, pp. 1070–1079.

[19] Y. W. Teh, Dirichlet process, in: Encyclopedia of machine learning, Springer, 2011, pp. 280–287.

[20] J. Sethuraman, A constructive definition of Dirichlet priors, Statistica sinica (1994) 639–650.

[21] C. M. Carvalho, H. F. Lopes, N. G. Polson, M. A. Taddy, et al., Particle learning for general mixtures, Bayesian Analysis 5 (4) (2010) 709–740.

[22] P. Fearnhead, Particle filters for mixture models with an unknown number of components, Statistics and Computing 14 (1) (2004) 11–21.

[23] S. Tong, D. Koller, Active learning for parameter estimation in Bayesian networks, in: NIPS, Vol. 13, 2000, pp. 647–653.

[24] N. Roy, A. McCallum, Toward optimal active learning through monte carlo estimation of error reduction, ICML, Williamstown (2001) 441–448.

[25] G. W. Taylor, G. E. Hinton, S. T. Roweis, Modeling human motion using binary latent variables, in: Advances in neural information processing systems, 2006, pp. 1345–1352.

[26] M. D. Zeiler, G. W. Taylor, N. F. Troje, G. E. Hinton, Modeling pigeon behavior using a Conditional Restricted Boltzmann Machine., in: ESANN, 2009.

[27] A.-r. Mohamed, G. Hinton, Phone recognition using Restricted Boltzmann Machines, in: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, IEEE, 2010, pp. 4354–4357.

[28] Z. Wu, E. S. Chng, H. Li, Conditional Restricted Boltzmann Machines for voice conversion, in: Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on, IEEE, 2013, pp. 104–108.

[29] G. W. Taylor, L. Sigal, D. J. Fleet, G. E. Hinton, Dynamical binary latent variable models for 3d human pose tracking, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 631–638.

[30] V. Mnih, H. Larochelle, G. E. Hinton, Conditional Restricted Boltzmann Machines for structured output prediction, arXiv preprint arXiv:1202.3748.

[31] J. Li, W. Zhang, Conditional Restricted Boltzmann Machines for cold start recommendations, arXiv preprint arXiv:1408.0096.

[32] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, Z. Yu, Sensor-based activity recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (6) (2012) 790–808.

[33] O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Communications Surveys & Tutorials 15 (3) (2013) 1192–1209.

[34] K. Altun, B. Barshan, Human activity recognition using inertial/magnetic sensor units, in: International Workshop on Human Behavior Understanding, Springer, 2010, pp. 38–51.

[35] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, M. Beigl, Actiserv: Activity recognition service for mobile phones, in: International Symposium on Wearable Computers (ISWC) 2010, IEEE, 2010, pp. 1–8.

[36] J. R. Kwapisz, G. M. Weiss, S. A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter 12 (2) (2011) 74–82.

[37] W. Xu, M. Zhang, A. A. Sawchuk, M. Sarrafzadeh, Robust human activity and sensor location corecognition via sparse signal representation, IEEE Transactions on Biomedical Engineering 59 (11) (2012) 3169–3176.

[38] C. Catal, S. Tufekci, E. Pirmit, G. Kocabag, On the use of ensemble of classifiers for accelerometer-based activity recognition, Applied Soft Computing 37 (2015) 1018–1022.

[39] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[40] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina, 2015, pp. 25–31.

[41] Y. Du, W. Wang, L. Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1110–1118.

[42] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, G. Taylor, Learning human identity from motion patterns, IEEE Access 4 (2016) 1810–1820.

[43] N. Y. Hammerla, S. Halloran, T. Ploetz, Deep, convolutional, and recurrent models for human activity recognition using wearables, arXiv preprint arXiv:1604.08880.

[44] T. Plötz, N. Y. Hammerla, P. Olivier, Feature learning for activity recognition in ubiquitous computing, in: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, Vol. 22, 2011, p. 1729.

[45] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, S. Krishnaswamy, Activity recognition with evolving data streams: A review, ACM Computing Surveys (CSUR) 51 (4) (2018) 71.

[46] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, S. Krishnaswamy, Adaptive mobile activity recognition system with evolving data streams, Neurocomputing 150 (2015) 304–317.

[47] T. Diethe, N. Twomey, P. A. Flach, Active transfer learning for activity recognition., in: ESANN, 2016.

[48] R. Salakhutdinov, J. B. Tenenbaum, A. Torralba, Learning with hierarchical-deep models, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1958–1971.

[49] D. A. Cohn, Z. Ghahramani, M. I. Jordan, Active learning with statistical models, Journal of artificial intelligence research.

[50] S. Vijayanarasimhan, K. Grauman, Multi-level active prediction of useful image annotations for recognition, in: Advances in Neural Information Processing Systems, 2009, pp. 1705–1712.

[51] S. Mohamad, A. Bouchachia, M. Sayed-Mouchaweh, A bi-criteria active learning algorithm for dynamic data streams, IEEE transactions on neural networks and learning systems 29 (1) (2018) 74–86.

[52] S. Mohamad, M. Sayed-Mouchaweh, A. Bouchachia, Active learning for classifying data streams with unknown number of classes, Neural Networks 98 (2018) 1–15.

[53] D. J. MacKay, Information-based objective functions for active data selection, Neural computation 4 (4) (1992) 590–604.

[54] Y. Freund, H. S. Seung, E. Shamir, N. Tishby, Selective sampling using the query by committee algorithm, Machine learning 28 (2-3) (1997) 133–168.

[55] N. Houlsby, F. Huszár, Z. Ghahramani, M. Lengyel, Bayesian active learning for classification and preference learning, arXiv preprint arXiv:1112.5745.

[56] I. Žliobaitė, A. Bifet, B. Pfahringer, G. Holmes, Active learning with drifting streaming data, IEEE transactions on neural networks and learning systems 25 (1) (2014) 27–39.

[57] M. West, Hyperparameter estimation in Dirichlet process mixture models, Duke University ISDS Discussion Paper# 92-A03, 1992.

[58] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, D. Roggen, The opportunity challenge: A benchmark database for on-body sensor-based activity recognition, Pattern Recognition Letters 34 (15) (2013) 2033–2042.

[59] P. Casale, O. Pujol, P. Radeva, Personalization and user verification in wearable systems using biometric walking patterns 16 (2012) 1–18.

[60] S. Mohamad, A. Bouchachia, M. Sayed-Mouchaweh, A non-parametric hierarchical clustering model, in: Evolving and Adaptive Intelligent Systems (EAIS), 2015 IEEE International Conference on, IEEE, 2015, pp. 1–7.

[61] J. D. McAuliffe, D. M. Blei, M. I. Jordan, Nonparametric empirical bayes for the Dirichlet process mixture model, Statistics and Computing 16 (1) (2006) 5–14.

[62] I. H. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2005.

[63] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527–1554.

[64] R. M. Neal, Bayesian mixture modeling, in: Maximum Entropy and Bayesian Methods, Springer, 1992, pp. 197–211.

[65] C. E. Rasmussen, The infinite gaussian mixture model., in: NIPS, Vol. 12, 1999, pp. 554–560.

[66] T. S. Ferguson, A Bayesian analysis of some nonparametric problems, The annals of statistics (1973) 209–230.

[67] D. Blackwell, J. B. MacQueen, Ferguson distributions via pólya urn schemes, The annals of statistics (1973) 353–355.

[68] J. Sethuraman, A constructive definition of Dirichlet priors, Statistica sinica (1994) 639–650.

[69] R. M. Neal, Markov chain sampling methods for Dirichlet process mixture models, Journal of computational and graphical statistics 9 (2) (2000) 249–265.

[70] P. J. Bickel, K. A. Doksum, Mathematical statistics: basic ideas and selected topics, Vol. 2, CRC Press, 2015.

[71] C. E. Antoniak, Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems, The annals of statistics (1974) 1152–1174.

Appendix A. Conditional Restricted Boltzmann Machine

CRBM [25] is a non-linear generative model for time-series that uses an undirect model with binary latent variables, **h**, connected to visible variables, **v**. Unlike Hidden Markov models (HMMs) which rely on a single discrete K-state multinomial, CRBM allows for distributed binary representations for its hidden states. For, example, to model N bits of information about the past history, HMMs require 2^N hidden states, while CRBM only needs N binary latent variables. Linear dynamical systems are models with distributed hidden state, but, they cannot model the complex non-linear dynamics in the high-dimensional sensory input.

CRBM is a temporal extension of restricted Boltzmann machines (RBM). Typically, RBM uses binary units for both visible and hidden variables. But the sensory input in our data is continuous; therefore, we use real-valued Gaussian input units. CRBM has a layer of visible units which resembles to autoregressive model and a layer of hidden units. The visible variables **v** and hidden variables **h** in the current time slice receive directed connections from the visible variables at the previous few time slices. Also, there are undirected connections between layers at the current time slice like in RBM. Figure A.6 shows a CRBM example with two layers, where the temporal order of the one at the bottom (r_1) is 2 and for the one in the top (r_2) is 1. CRBM defines a joint probability distribution over **v** and **h**, conditional on the past n observations and model parameters Φ:

$$\begin{aligned} & p(\boldsymbol{v}, \boldsymbol{h} | \{\boldsymbol{v}\}_{t-n}^{t-1}, \Phi) \propto \exp(-E(\boldsymbol{v}, \boldsymbol{h} | \{\boldsymbol{v}\}_{t-n}^{t-1}, \Phi)) \\ & E(\boldsymbol{v}, \boldsymbol{h} | \{\boldsymbol{v}\}_{t-n}^{t-1}, \Phi) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j h_j b_j - \sum_{ij} \phi_{ij} \frac{v_i}{\sigma_i} h_j \end{aligned} \tag{A.1}$$

where σ_i is the standard deviation of the Gaussian noise for visible unit i. Like in [25], it is set to one after rescaling the data to have zero mean and unit variance. The dynamic biases, b_i, b_j, are affine functions of the past n observations. The parameter ϕ_{ij} is a weight between elements v_i and h_j. The undirected connections between the hidden and visible variables in the current layer (time

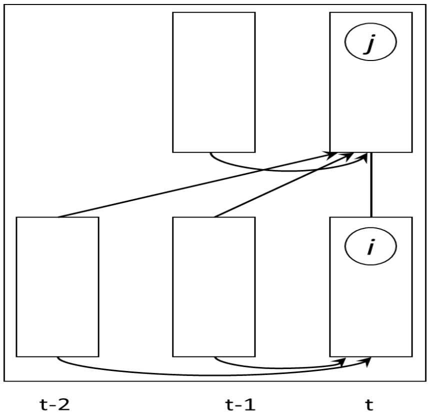


Figure. A.6 Feature Extractor Architecture ($r_1 = 2$, $r_2 = 1$)

Table. A.15 CRBM's parameters

CRBM layers	input dimension	output dimension	temporal order
layer 1	113	150	8
layer 2	150	200	0
layer 3	200	10	0

t) makes the inference easy because the hidden units become conditionally independent when the visible units are observed. The training is, therefore, easily done by minimizing contrastive divergence (for more details see [25]).

A crucial characteristic of CRBM is that we can add layers like in Deep Belief Networks [63]. All layers in the CRBM architecture are trained similarly but sequentially. As more layers are added, CRBM can model higher-level features. In this paper, we use only two layers to retain the low-level inter-features correlations at a lower training computational cost.

Before running experiments, we set up CRBM by training it on the Opp data. To ensure the independence between the online learning and feature extraction, the CRBM model for each subject is built from unlabelled data of all other subjects (all subjects except the one on which the current online learning is evaluated). We study the impact of different CRBM's parameter settings. The number of layers is fixed to 3 with the dimension of the last layer is set to 10

($dim3 = 10$). The effect of the CRBM parameters effect (number of layers, dimensions and temporal orders) is studied and the parameters that give the best performance are chosen. The first and second layers' dimensions are set to $dim1 = 150$ and $dim2 = 200$ respectively, the temporal orders are set to $r1 = 8$, $r2 = 0$ and $r3 = 0$ (see Tab. A.15). In order to demonstrate what CRBM has learned about the structure of the data, we feed the trained CRBM with a data segment consisting of 22000 samples from subject 3 (S3) and plot the results.

Figure A.7 is a gray-scale image showing the probability of the binary features extracted by the first layer. In Figure A.8, the features extracted by the second layer is shown with a ribbon of different colours which illustrates some class labels of the activities through time. We can notice a regular output pattern for each activity. It can also be seen that the regularity of the features obtained by the second layer (Fig. A.8) is sharper and less noisy than those obtained by the first layer (Fig. A.7). Hence, features become more discriminative. To visualize the data samples in the features space, we set the last layer's dimensions to 2 and plot the output in Fig. A.9. The data samples representing *standing* activity often overlap with the other activities especially *walking*. A potential explanation is that most transitions are from *standing* to *walking*. Besides, the gesture activities (not plotted) are usually performed while the user is standing. Thus, the *standing* activity region in the feature space gets expanded towards other activities' regions.

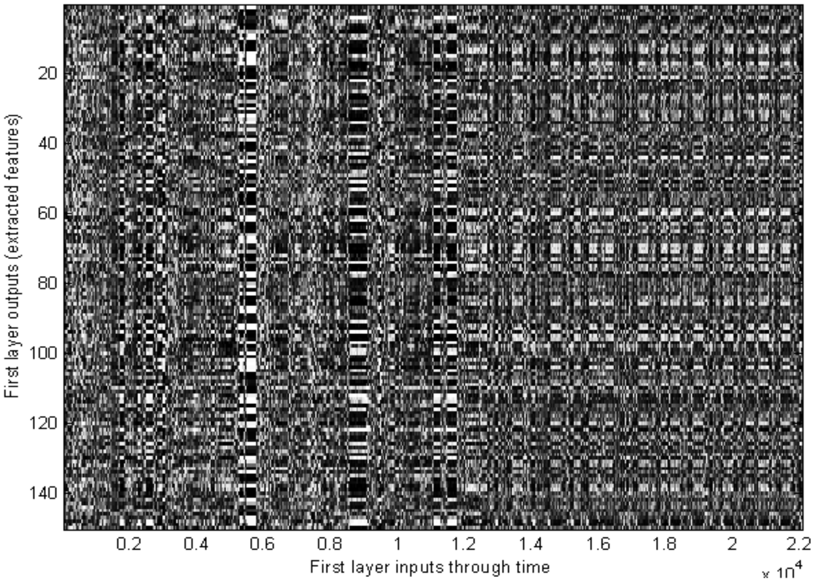


Figure. A.7 Layer 1

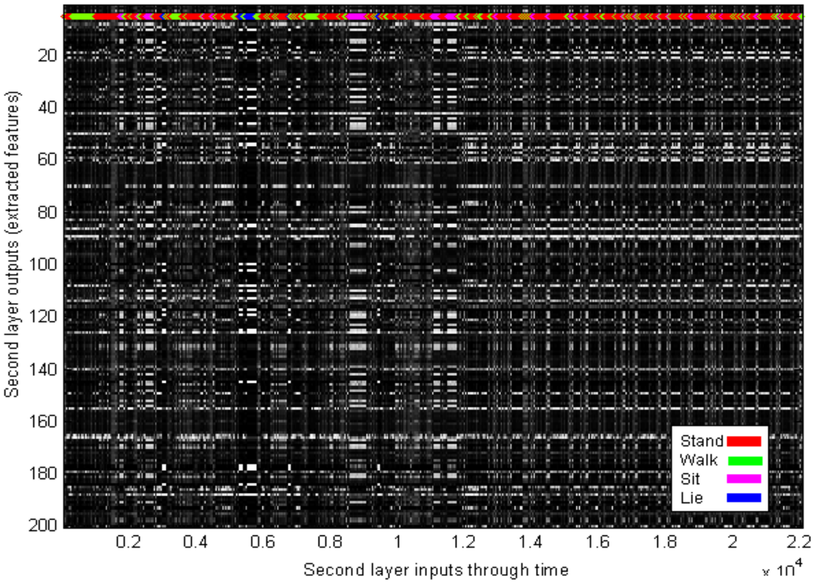


Figure. A.8 Layer 2

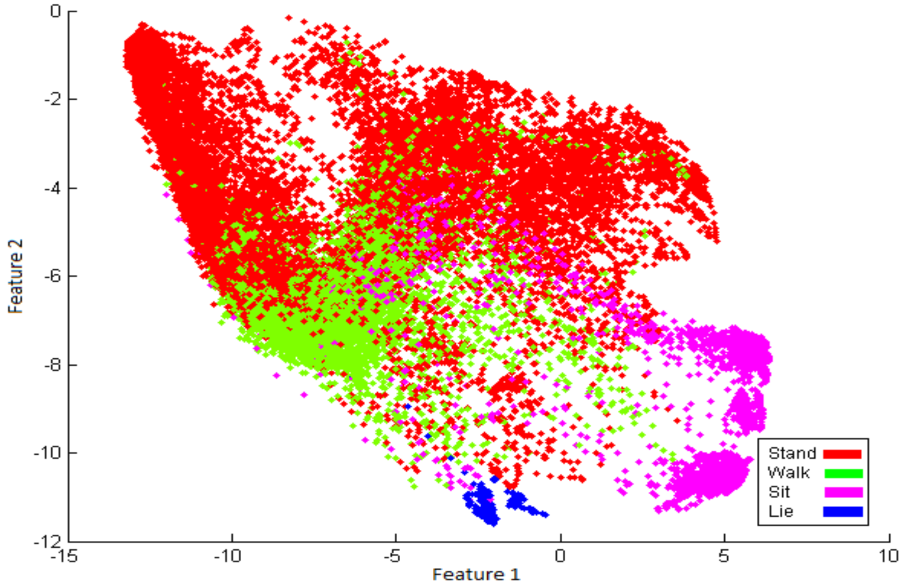
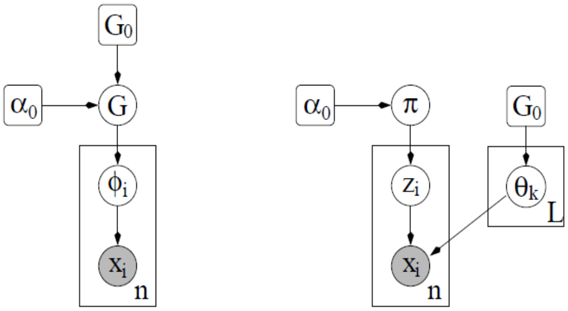


Figure. A.9 Layer 3

Appendix B. Dirichlet process

DP is one of the most popular prior used in Bayesian non-parametric modelling. It was first use by the machine learning community in [64, 65]. In general, stochastic process is probability distribution over a space of paths which describe the evolution of some random value over time. DP is a family of stochastic processes whose paths are probability distributions. It can be seen as an infinite-dimensional generalization of Dirichlet distribution. In the literature, DP has been constructed with different ways, the most well-known constructions are: infinite mixture model [65], distribution over distribution [66], Polya-urn scheme [67] and stick-breaking [68]. For more details, interested reader is referred to [19].

Figure B.10 shows two graphical models, DP mixture model and the finite mixture model with a number of clusters L which becomes an infinite mixture model when L goes to ∞. Infinite mixture model is simply a generalization



(a) DP mixture model (b) Finite mixture model

Figure. B.10 Graphical model

of the finite mixture model, where DP prior with infinite parameters is used instead of Dirichlet distribution prior with fixed number of parameters. The finite mixture model can be represented by the following equations:

$$\begin{aligned}
\boldsymbol{\pi}|\alpha_0 &\sim Dirichlet(\alpha_0/L, ..., \alpha_0/L) \\
z_i|\boldsymbol{\pi} &\sim Discrete(\pi_1, ..., \pi_L) \\
\boldsymbol{\theta_k}|G_0 &\sim G_0 \\
\boldsymbol{x_i}|z_i, \boldsymbol{\theta} &\sim F(\boldsymbol{\theta_{z_i}}) \qquad \text{(B.1)}
\end{aligned}$$

$F(\boldsymbol{\theta_{z_i}})$ denotes the distribution of the observation $\boldsymbol{x_i}$ given $\boldsymbol{\theta_{zi}}$, where $\boldsymbol{\theta_{zi}}$ is the parameter vector associated with component z_i. Here z_i indicates which latent cluster is associated with observation $\boldsymbol{x_i}$. Indicator z_i is drawn from a discrete distribution governed by parameter $\boldsymbol{\pi}$ drawn from Dirichlet distribution parametrized by α_0. We can simply say that $\boldsymbol{x_i}$ is distributed according to a mixture of components drawn from prior distribution G_0 and picked with probability given by the vector of mixing proportions $\boldsymbol{\pi}$. The model represented by Eq.(B.1) above is a finite mixture model, where L is the fixed number of parameters (components). The infinite mixture model can be derived by letting $L \rightarrow \infty$, then $\boldsymbol{\pi}$ can be represented as an infinite mixing proportion distributed according to stick-breaking process $GEM(\alpha_0)$ [68]. Thus, Eq.(B.1) can be equivalently expressed according to the graphical representation as follows:

$$\begin{aligned} G|\alpha_0, G_0 &\sim DP(G_0, \alpha_0) \\ \boldsymbol{\theta_i}|G &\sim G \\ \boldsymbol{x_i}|\boldsymbol{\theta_i} &\sim F(\boldsymbol{\theta_i}) \end{aligned} \tag{B.2}$$

where $G = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta_i}}$ is drawn from the DP prior. $\delta_{\boldsymbol{\theta_i}}$ is a Dirac delta function centred at $\boldsymbol{\theta_i}$. Technically, DP is a distribution over distributions [66], where $DP(G_0, \alpha)$, is parametrized by the base distribution G_0, and the concentration parameter α. Since DP is distribution over distributions, a draw G from it is a distribution. Thus, we can sample $\boldsymbol{\theta_i}$ from G. Back to Eq.(B.1), by integrating over the mixing proportion $\boldsymbol{\pi}$, we can write the prior for z_i as conditional probability of the following form [69]:

$$p(z_i = c|z_1, ..., z_{i-1}) = \frac{n_c^{-i} + \alpha_0/L}{i - 1 + \alpha_0} \tag{B.3}$$

where n_c^{-i} is the number of data samples excluding $\boldsymbol{x}_i$ that are assigned to component c. By letting L go to infinity we get the following equations:

$$\begin{aligned} p(z_i = c|z_1, ..., z_{i-1}) &\rightarrow \frac{n_c^{-i}}{i - 1 + \alpha_0} \\ p(z_i \neq z_j \, for \, all \, j < i|z_1, ..., z_{i-1}) &\rightarrow \frac{\alpha_0}{i - 1 + \alpha_0} \end{aligned} \tag{B.4}$$

For an observation $\boldsymbol{x_i}$ with $z_i \neq z_j \, for \, all \, j < i$, a new component is created with indicator $z_i = c_{new}$. For more details about the process of obtaining the prior distribution, the reader is referred to [69].

Appendix C. Notations

Symbol	**Description**
$\boldsymbol{x_t}$	data input at time t
y_t	data label at time t
$\boldsymbol{\pi}$	a draw from stick-breaking process
α	stick-breaking process hyper-parameter
G_0	is a Normal-Inverse-Wishart distribution
z_t	an indicator of the component (cluster) generating x_t
D_t	the previously seen data samples up to time t along with their labels if provided
$n_{i,j,t}$	is the number of data samples labeled i and assigned to component j at time t
$\boldsymbol{\theta}$	a component parameters
$\boldsymbol{s}_{j,t}$	is the sufficient statistics (i.e., mean and scatter matrix) of component j at t
$\boldsymbol{su}_{j,t}$	mean of data samples in component j at time t
$\boldsymbol{cu}_{j,t}$	scatter matrix of data samples in component j at time t
H_t	a state vector that summarizes the data seen up to time t
m_t	the number of components at time t
P	maximum allowed number of particles
$w_t^{(i)}$	weight of particle i at time t
λ	memory factor
C_t	the set of the labels of all existing classes at time t
Ω	set of variables that can be fully random or include some random elements
L	loss function
R	Risk
$\hat{R}$	expected Risk
Q	set of binary variable
ψ	model parameters
X	set of data samples
Y	set of data labels
$\hat{\Delta}$	difference between the current risk and the current expected risk
f_t	number of labelled instances at time t
X_{L_t}	set of labelled data samples seen up to time t
b_t	budget spent up to time t
X_t	set of data samples seen up to time t
Bd	a constant represents the maximum allowed budget
Lab_t	a binary variable to indicate whether the data sample at time t is labelled or not

Appendix D. Computation of Eq. (1)

$$p(y_t|\boldsymbol{x}_t, D_{t-1}) \propto p(\boldsymbol{x}_t|y_t, D_{t-1})p(y_t|D_{t-1}) \tag{D.1}$$

$$p(y_t|D_{t-1}) \propto \begin{cases} n_{y_t,:,t} & y_t \text{ is an existing class} \\ \alpha_2 & y_t \text{ is a new class} \end{cases} \tag{D.2}$$

where $n_{y_t,:,t}$ is the number of data samples labeled y_t at time t regardless of the components. The ':' denotes all the components.

$$p(\boldsymbol{x}_t|y_t, D_{t-1}) = \sum_{z_{1:t-1}} p(\boldsymbol{x}_t|y_t, z_{1:t-1}, D_{t-1})p(z_{1:t-1}|y_t, D_{t-1}) \tag{D.3}$$

$$p(\boldsymbol{x}_t|y_t, z_{1:t-1}, D_{t-1}) = \sum_{z_t} p(\boldsymbol{x}_t|y_t, z_t, z_{1:t-1}, D_{t-1})p(z_t|y_t, z_{1:t-1}, D_{t-1}) \tag{D.4}$$

$$p(z_{1:t-1}|y_t, D_{t-1}) \propto p(y_t|D_{t-1})p(z_{1:t-1}|D_{t-1}) \tag{D.5}$$

We use $z_{1:t}$ to denote the sequence $\{z_1, z_2, ..., z_t\}$. The first term of Eq. (D.4) can be computed as follows:

$$p(\boldsymbol{x}_t|y_t, z_t, z_{1:t-1}, D_{t-1}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t|\boldsymbol{\theta}, z_t)p(\boldsymbol{\theta}|z_t, z_{1:t-1}, D_{t-1}) \tag{D.6}$$

If z_t refers to a new component, $p(\boldsymbol{\theta}|z_t, z_{1:t-1}, D_{t-1})$ becomes equivalent to the prior distribution $p(\boldsymbol{\theta}|G_0)$. Otherwise, z_t refers to an already existing component. Then, $p(\boldsymbol{\theta}|z_t, z_{1:t-1}, D_{t-1})$ becomes equivalent to $p(\boldsymbol{\theta}|\boldsymbol{s}_{z_t,t-1}, n_{z_t,t-1}, z_{1:t-1})$, where $\boldsymbol{s}_{z_t,t-1} = \{\boldsymbol{su}_{z_t,t-1}, \boldsymbol{sc}_{z_t,t-1}\}$ is the sufficient statistics (i.e., mean and scatter matrix respectively) defined as.

$$\begin{aligned} \boldsymbol{su}_{z_t,t-1}(z_{1:t-1}) &= \frac{\sum_{z_i=z_t, i<t} \boldsymbol{x}_i}{n_{z_t,t-1}} \\ \boldsymbol{sc}_{z_t,t-1}(z_{1:t-1}) &= \sum_{z_i=z_t, i<t} (\boldsymbol{x}_i - \boldsymbol{su}_{z_t,t-1})(\boldsymbol{x}_i - \boldsymbol{su}_{z_t,t-1})^T \end{aligned} \tag{D.7}$$

where $n_{z_t,t-1}$ is the number of data samples which have been assigned to component z_t until time $t-1$. Equation (D.6) can be solved given the sufficient statistics, the past assignments and the model hyper-parameters. More details can be found in Appendix E. The second term of Eq. (D.4) can be written as follows:

$$p(z_t|y_t, z_{1:t-1}, D_{t-1}) = p(z_t|\{z_i\}_{y_i=y_t, i<t}) \tag{D.8}$$

Similar to Eq. (D.6), the solution is as follows:

$$p(z_t|\{z_i\}_{y_i=y_t, i<t}) \propto \begin{cases} n_{y_t,z_t,t} & z_t \text{ is an existing component} \\ \alpha_1 & z_t \text{ is a new component} \end{cases} \tag{D.9}$$

where $n_{y_t,z_t,t}$ is the number of data samples labeled y_t and assigned to component z_t at time t.

The second term in Eq. (D.5), $p(z_{1:t-1}|D_{t-1})$, is the probability of the different configurations. Such configurations determine the different statistics represented by H_t. Thus, $p(H_{t-1}|D_{t-1})$ has the same probability as the posterior $p(z_{1:t-1}|D_{t-1})$. $p(H_{t-1}|D_{t-1})$ is outlined and developed in Eq. 2.

Appendix E. Computation of Eq. (D.6)

- If z_t refers to a new component:

$$p(\boldsymbol{x}_t|y_t, z_{1:t}, D_{t-1}) = \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t|\boldsymbol{\theta})p(\boldsymbol{\theta}|G_0) = t_{v_1}(\boldsymbol{x}_t|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \tag{E.1}$$

where t refer to student's t-distribution resulting from using a conjugate prior (i.e., the Normal Inverse Wishart prior) over the normal distribution parameter $\boldsymbol{\theta}$.

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 \tag{E.2}$$

$$\boldsymbol{\Sigma}_1 = \frac{\boldsymbol{\Sigma}_0(k_0+1)}{k_0(v_0-d+1)} \tag{E.3}$$

$$v_1 = v_0 - d + 1 \tag{E.4}$$

where d is the dimension of the data.

- If z_t refers to an existing component, then:

$$\begin{aligned} p(\boldsymbol{x}_t|y_t, z_{1:t}, D_{t-1}) &= \int_{\boldsymbol{\theta}} p(\boldsymbol{x}_t|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{s}_{z_t,t-1}(z_{1:t-1}), n_{z_t,t-1}) \\ &= t_{v_2}(\boldsymbol{x}_t|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned} \tag{E.5}$$

where:

$$\boldsymbol{\mu}_2 = \frac{k_0}{k_0 + n_{z_t,t-1}}\boldsymbol{\mu}_0 + \frac{n_{z_t,t-1}}{k_0 + n_{z_t,t-1}}\boldsymbol{su}_{z_t,t-1} \tag{E.6}$$

$$\begin{aligned} \boldsymbol{\Sigma}_2 = \frac{1}{(k_0 + n_{z_t,t-1})(v_0 + n_{z_t,t-1} - d + 1)} \big(\boldsymbol{\Sigma}_0 + \boldsymbol{sc}_{z_t,t-1} + \\ \frac{k_0 n_{z_t,t-1}}{k_0 + n_{z_t,t-1}}(\boldsymbol{su}_{z_t,t-1} - \boldsymbol{\mu}_0)(\boldsymbol{su}_{z_t,t-1} - \boldsymbol{\mu}_0)^T\big) \\ (k_0 + n_{z_t,t-1} + 1) \end{aligned} \tag{E.7}$$

and

$$v_2 = v_0 + n_{z_t,t-1} - d + 1 \tag{E.8}$$

Appendix F. Computation of Eq. (19)

After marginalizing out the Gaussian and the stick-breaking components, $\boldsymbol{\psi}_t$ becomes equivalent to $\boldsymbol{z}_{1:t}$. Thus, the discrepancy between the current risk and the current expected risk can be written as follows:

$$\hat{\Delta}(\boldsymbol{x}_t, y_t|D_{t-1}, q_t) = R(p(z_{1:t}|D_{t-1}, \boldsymbol{x}_t), \hat{z}_{1:t}) - \hat{R}(p(z_{1:t}|D_{t-1}, \boldsymbol{x}_t, y_t), \hat{z}_{1:t}; q_t) \tag{F.1}$$

$$\begin{aligned} R(p(z_{1:t}|D_{t-1},\boldsymbol{x}_t),\hat{z}_{1:t}) &= E_{z_{1:t}\sim p(z_{1:t}|D_{t-1},\boldsymbol{x}_t)}[L(z_{1:t},\hat{z}_{1:t})] \\ &= \sum_{z_{1:t}} p(z_{1:t}|D_{t-1},\boldsymbol{x}_t)L(z_{1:t},\hat{z}_{1:t}) \end{aligned} \tag{F.2}$$

Given that the data instance at time t is queried ($q_t = 1$), the current expected risk can presented as follows:

$$\hat{R}(p(z_{1:t}|D_{t-1},\boldsymbol{x}_t,y_t),\hat{z}_{1:t};q_t=1) = \sum_{y_t} p(y_t|D_{t-1},\boldsymbol{x}_t)R(p(z_{1:t}|D_{t-1},\boldsymbol{x}_t,y_t),\hat{z}_{1:t}) \tag{F.3}$$

To compute Eq. (F.1), both Eq. (F.2) and Eq. (F.3) must be solved. Given that the loss function in Eq. (F.2) is solved, the computation of Eq. (F.2) and Eq. (F.3) is straightforward. Thus, we start by the loss function which can be written as follows:

$$L(z_{1:t},\hat{z}_{1:t}) = A - B \tag{F.4}$$

where:

$$A = \sum_{y}\int_{\boldsymbol{x}} p(\boldsymbol{x},y|D_t,z_{1:t})\log\big(p(\boldsymbol{x},y|D_t,z_{1:t})\big)d\boldsymbol{x} \tag{F.5}$$

$$B = \sum_{y}\int_{\boldsymbol{x}} p(\boldsymbol{x},y|D_t,z_{1:t})\log\big(p(\boldsymbol{x},y|D_t,\hat{z}_{1:t})\big)d\boldsymbol{x} \tag{F.6}$$

$$A = A_1 + A_2 \tag{F.7}$$

where

$$\begin{aligned} A_1 = \sum_{y}\log\big(p(y|D_t,z_{1:t})\big)p(y|D_t,z_{1:t})\sum_{z}p(z|y,z_{1:t},D_t) \\ \int_{\boldsymbol{x}}\int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z,\boldsymbol{\theta})p(\boldsymbol{\theta}|z,D_t,z_{1:t})d\boldsymbol{\theta}d\boldsymbol{x} \end{aligned} \tag{F.8}$$

The integral over $\boldsymbol{\theta}$ leads to a t-student distribution as shown in Eq. (D.6). Hence,

$$A_1 = \sum_y \log\big(p(y|D_t, z_{1:t})\big)p(y|D_t, z_{1:t}) \tag{F.9}$$

where the terms of Eq. (F.9) are already computed in Eq. (D.2).

$$\begin{aligned} A_2 = \sum_y p(y|D_t, z_{1:t}) \sum_z p(z|y, z_{1:t}, D_t) \int_{\boldsymbol{x}} \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) \\ p(\boldsymbol{\theta}|z, D_t, z_{1:t}) d\boldsymbol{\theta} \log\Bigg(\sum_z p(z|y, z_{1:t}, D_t) \\ \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) p(\boldsymbol{\theta}|z, D_t, z_{1:t}) d\boldsymbol{\theta}\Bigg) d\boldsymbol{x} \end{aligned} \tag{F.10}$$

Similarly, we compute B:

$$B = B_1 + B_2 \tag{F.11}$$

$$\begin{aligned} B_1 = \sum_y \log\big(p(y|D_t, \hat{z}_{1:t})\big)p(y|D_t, z_{1:t}) \sum_z p(z|y, z_{1:t}, D_t) \\ \int_{\boldsymbol{x}} \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) p(\boldsymbol{\theta}|z, D_t, z_{1:t}) d\boldsymbol{\theta} d\boldsymbol{x} \end{aligned} \tag{F.12}$$

$$B_1 = \sum_y \log\big(p(y|D_t, \hat{z}_{1:t})\big)p(y|D_t, z_{1:t}) \tag{F.13}$$

$$\begin{aligned} B_2 = \sum_y p(y|D_t, z_{1:t}) \sum_z p(z|y, z_{1:t}, D_t) \int_{\boldsymbol{x}} \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) \\ p(\boldsymbol{\theta}|z, D_t, z_{1:t}) d\boldsymbol{\theta} \log\Bigg(\sum_z p(z|y, \hat{z}_{1:t}, D_t) \\ \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) p(\boldsymbol{\theta}|z, D_t, \hat{z}_{1:t}) d\boldsymbol{\theta}\Bigg) d\boldsymbol{x} \end{aligned} \tag{F.14}$$

Given that

$$
\begin{aligned}
Q_1(\boldsymbol{x}) &= \sum_z p(z|y, z_{1:t}, D_t) \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) p(\boldsymbol{\theta}|z, D_t, z_{1:t}) d\boldsymbol{\theta} \\
Q_2(\boldsymbol{x}) &= \sum_z p(z|y, \hat{z}_{1:t}, D_t) \int_{\boldsymbol{\theta}} p(\boldsymbol{x}|z, \boldsymbol{\theta}) p(\boldsymbol{\theta}|z, D_t, \hat{z}_{1:t}) d\boldsymbol{\theta} \\
g(\boldsymbol{x}) &= \log \frac{Q_1(\boldsymbol{x})}{Q_2(\boldsymbol{x})} \qquad \text{(F.15)}
\end{aligned}
$$

$A_2 - B_2$ can be written as follows:

$$
\begin{aligned}
&A_2 - B_2 = \sum_y p(y|D_t, z_{1:t}) \int_{\boldsymbol{x}} Q_1(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} = \\
&\sum_y p(y|D_t, z_{1:t}) \sum_z \int_{\boldsymbol{x}} p(z|y, z_{1:t}, D_t) t(\boldsymbol{x}|z, D_t, z_{1:t}) g(\boldsymbol{x}) d\boldsymbol{x} \qquad \text{(F.16)}
\end{aligned}
$$

where $t(\boldsymbol{x}|.)$ refers to student's t-distribution. We can approximate each term in this sum with a second order Taylor series expansion of $g(\boldsymbol{x})$ around the means $\boldsymbol{\mu}_z \equiv \boldsymbol{\mu}_{(z, z_1:t, D_t)}$ of the student's t-distribution's components:

$$
g(\boldsymbol{x}) \approx \hat{g}_{\boldsymbol{\mu}_z}(\boldsymbol{x}) = g(\boldsymbol{\mu}_z) + \nabla g(\boldsymbol{\mu}_z)(\boldsymbol{x} - \boldsymbol{\mu}_z) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_z)^T \nabla^2 g(\boldsymbol{\mu}_z)(\boldsymbol{x} - \boldsymbol{\mu}_z) \qquad \text{(F.17)}
$$

where ∇g and $\nabla^2 g$ are the gradient and the Hessian matrix of the second derivatives. Hence, Eq. (F.16) can be written as follows:

$$
A_2 - B_2 = \sum_y p(y|D_t, z_{1:t}) \sum_z p(z|y, z_{1:t}, D_t) \Big(g(\boldsymbol{\mu}_z) + \frac{1}{2} tr\big(\nabla^2 g(\boldsymbol{\mu}_z) \frac{v_z}{v_z - 2} \boldsymbol{\Sigma}_z\big) \Big) \qquad \text{(F.18)}
$$

where v_z and $\boldsymbol{\Sigma}_z$ are the degree of freedom and the covariance matrix of the student's t-distribution's component determined by z. This approximation is known as the multivariate delta method for moments [70]. Finally, the loss function in Eq. (F.4) can be written as follows:

$$\begin{aligned} L(z_{1:t}, \hat{z}_{1:t}) = \sum_{y} p(y|D_t, z_{1:t}) \Bigg[\log \frac{p(y|D_t, z_{1:t})}{p(y|D_t, \hat{z}_{1:t})} + \\ \sum_{z} p(z|y, z_{1:t}, D_t) \bigg(g(\boldsymbol{\mu}_z) + \frac{1}{2} tr\big(\nabla^2 g(\boldsymbol{\mu}_z) \frac{v_z}{v_z - 2} \boldsymbol{\Sigma}_z\big) \bigg) \Bigg]. \end{aligned} \tag{F.19}$$

The current risk in Eq. (F.2) can be easily computed by replacing the loss function with its solution in Eq. (F.19). Thus, the current expected risk in Eq. (F.3) can be computed by replacing the current risk with its solution. Hence, the discrepancy between the current risk and the current expected risk in Eq. (F.1) is solved.

Appendix G. Sampling precision hyper-parameters

The authors in [57] show that the precision parameter in a DP mixture model α is conditionally independent of the data given the number of distinct components m and the size of the data, $n = \boldsymbol{e}^T \boldsymbol{ne}$. Let $\alpha \sim G(a, b)$, a gamma prior with shape a and scale b which are both fixed to 1. The posterior distribution of α can be written as follows:

$$p(\alpha|m, n) \propto p(\alpha) p(m|\alpha, n) \tag{G.1}$$

According to [71], the likelihood in Eq. (G.1) may be written as:

$$p(m|\alpha, n) = c_n(m) n! \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \tag{G.2}$$

where $c_n(m) = p(m|\alpha = 1, n)$ does not involve α. In this case, Eq. (G.1) can be expressed as mixture of two gamma posteriors [57].

$$(\alpha|\eta, m) \sim \pi_\eta G(a + m, b - \log(\eta)) + (1 - \pi_\eta) G(a + m - 1, b - \log(\eta)) \tag{G.3}$$

$$(\eta|\alpha, m) \sim B(\alpha + 1, n) \tag{G.4}$$

$$\frac{\pi_\eta}{1 - \pi_\eta} = \frac{a + m - 1}{n(b - \log(\eta))}$$

where B denotes the beta distribution, η is an auxiliary variable used in the sampling. To infer the distribution over α, Gibbs sampling iterations go as follows; First η is sampled from Eq. (G.4) conditional on the most recent value m and α. Second, α is sampled from Eq. (G.3) conditional on the already sampled η and the same m. The number of components can be deduced from the configuration variables.

To apply this sampling approach online on OSC model, we must sample for each class-specific mixture model, c, its corresponding, precision parameter α_{1c}, number of components m_{1c} and auxiliary variable η_{1c}. The set of parameters related to the class distribution, α_2, m_2 and η_2 must be sampled too. Furthermore, sampling must be done online. That is, once the data is processed, it is discarded. Thus, unlike offline Gibbs sampling, one set of precision parameters $\{\alpha_2^{new}, \alpha_{11}^{new}, ..., \alpha_{1m_2}^{new}\}$ is sampled in each iteration. The precision parameters are independent given the class label. Hence, they can be sampled independently using the same sampling routine described in [57].

Algorithm 2 Precision Parameters Sampling

```
function HypSamp($\{H_t^i, w_t^i\}_{i=1}^P, \{\alpha_2, \alpha_{1i}\}_{i=1}^{m_2} a, b$)
    Sample a particle: $h \sim \sum_{i=1}^P w_t^{(i)} \delta(H_t - H_t^{(i)})$
    Derive $\{m_2, m_{11}, ..., m_{1m_2}\}$ from $h$
    Sample $\{\eta_2, \eta_{11}, ..., \eta_{1m_2}\}$ $\big($Eq. (G.4)$\big)$
    Sample $\{\alpha_2^{new}, \alpha_{11}^{new}, ..., \alpha_{1m_2}^{new}\}$ $\big($Eq. (G.3)$\big)$
    Return $\{\alpha_2^{new}, \alpha_{11}^{new}, ..., \alpha_{1m_2}^{new}\}$
end function
```