

Exploring Concept Drift using Interactive Simulations

Jeremiah Smith, Naranker Dulay

Imperial College London,
{jeremiah,n.dulay}@imperial.ac.uk

Máté Attila Tóth., Oliver Amft

TU Eindhoven,
{a.m,amft}@tue.nl

Yanxia Zhang

Lancaster University,
yazhang@lancaster.ac.uk

Abstract—In machine learning, concept drift can cause the optimal solution to a given problem to change as time passes, leading to less accurate predictions. Concept drift can be sudden, gradual or reoccurring. Understanding the consequences of concept drift is particularly important in human-centric applications where changes in the underlying data and environment are common and unexpected. In order to gain a better understanding of the adverse effects of different types of concept drift on learners, we propose a novel simulation tool that is able to incrementally generate datasets with customisable concept drift by interacting with a human in a game-like setting. We illustrate our approach by generating and analysing concept drift simulations inspired by body-sensor based long-term activity recognition. Our initial results show that current unsupervised adaptation techniques can be caught in cyclic mislabelling and that a hybrid solution that is self-calibrating and semi-supervised is more robust than any of the two taken separately for this example.

Keywords—Machine Learning, Concept Drift, Activity Recognition, Semi-supervised Learning, Unsupervised Learning, Adaptive Learners

I. INTRODUCTION

Concept drift is a common occurrence in human-centric machine learning tasks. For example, concept drift has been shown to be present when trying to predict the sensitivity of a pathogen to antibiotics given patient samples and demographic data [13] (retrospectively found to be correlated with seasons, not present in the initial feature set), and in recommendation systems as a user's interests change [5].

In order to design robust machine learning systems for human-centric applications it is important to understand the consequences of concept drift on learner performance. A key requirement is to collect and analyse datasets over long periods of time. For example, the *PLCouple1* dataset collected at the MIT PlaceLab [8] provides 100 hours of annotated data, recorded over 9 days. Although this is substantially longer than most activity recognition data sets, 9 days of data is unable to accurately capture slow trends such as a change in a user's motor strategies due to ageing for instance. Moreover, given an arbitrary dataset, it is difficult to determine the type of concept drift without making any assumptions, for instance, how can we differentiate gradual drift versus noise, or overlapping gradual drifts on a finite dataset?

In this paper we propose a novel approach to generating datasets with concept drift using interactive simulations that can schedule and overlap different types of concept drift. The real-time output of simulations is visualised to show the effects of concept drift on different instances of learners. We are aware of three tools that can generate datasets with customisable drift [7] [9] [10]. The key difference between our tool and theirs is that the generated data and learning is done online using a human subject and aims to account for the variability intrinsic to many human-centric applications.

We illustrate our approach by considering body-sensor based long-term activity recognition, which is susceptible to concept drift due to sensor displacement [11]. Our initial results for this application show that current unsupervised adaptation techniques can be caught in cyclic mislabelling and that a hybrid solution that is self-calibrating and semi-supervised is more robust than any of the two taken separately.

The paper is structured as follows: Section II briefly summarises background work in concept drift, concept drift in activity recognition and concept drift simulation tools. Section III details our approach to concept drift simulation. Section IV illustrates how our tool has been used to generate a set of simulations related to long-term activity recognition. Section V summarises our findings for the aforementioned experiments and points to future work.

II. BACKGROUND

We first introduce concept drift. We then survey three concept drift simulation tools currently available and finally we investigate the state-of-the-art solutions to concept drift in activity recognition.

A. Concept drift

Concept drift refers to a change in the statistical properties of a target variable (the *concept* we are interested in predicting). For example, in classification tasks, this could be due to a change in the prior probabilities for a class as time passes. If a concept drift eventually alters the optimal solution for a given problem, performance of a learner is susceptible to fall as well if the drift is unaccounted for. The literature usually distinguishes between three concept drift

categories: *gradual* or *incremental drift*, *sudden* or *abrupt* and *recurring* drifts [14].

Intuitively, gradual concept drifts will happen over a relatively larger number of testing (training) instances than a sudden drift, while recurring concept drifts will oscillate between different testing (training) distributions, periodically coming back to a previously experienced one. A classification problem can be defined as assigning a class $c_i, i = 1 \dots K$ to a data point $x \in \mathbb{R}^n$ in n dimensional feature space. In this case, a learning problem can be expressed in terms of $P(c_i)$ the prior probability for each class and the class-conditional probability density function $p_i(x|c_j)$ for each class, for which we aim at finding the classifier \mathbf{C} that makes the least number of labelling errors. Concept drift can then occur because the class priors $P(c_i)$ change, or because the pdfs $p_i(x|c_j)$ change. Changes in posterior probabilities $P(c_j|x)$ and unconditional pdfs $p(x)$ can be rewritten in term of the above.

In practice it is usually not vital to know the exact location and cause of drift, rather we are interested in being able to adapt to it and most current solutions focus on using adaptive classifiers. A recent survey paper summarises the various strategies [15] to counter concept drift and the choice of strategy usually depends on the amount of labelled data available as well as the nature of the drift. In the example of body-sensor based long-term activity recognition, we are constrained by practical considerations such as the impossibility to periodically ask for a large set of consecutive labels, thus, for instance, sliding training window methods would not be appropriate for such an application.

B. Concept drift simulation toolsets

Our concept drift simulation tool generates datasets by taking advantage of human input via an interactive experiment as described in the next section. It is designed to run multiple learners in parallel and provides a simple way to visualise the behaviours and effects of planned drifts as they happen. In contrast, other toolsets produce stand-alone datasets are non-interactive and designed for offline processing or simulation. We are currently aware of three frameworks for generating datasets with concept drift.

The publicly available framework by Narasimhamurthy [10] is the most analytical of the three, and offers the most flexibility in the type of output data. It generates datasets by proportionally mixing arbitrary user defined distributions. Their approach is entirely defined by having a set of data sources with known class-conditional probability density functions $p_i(x|c_j), x \in \mathbb{R}^n$ with prior probabilities $P_i(c_j)$ for source i and class j . Assuming K sources, a data point at time t is then sampled using the mixing proportions $v_i(t) \in [0, 1]$ such that $P(c_j, t) = \sum_{i=1}^K v_i(t)P_i(c_j)$ and $P(x|c_j, t) = \sum_{i=1}^K v_i(t)p_i(x|c_j)$. Our tool differs from this framework in two main ways. First, it uses a simulation type approach that avoids the need to know, and thus encode, the

underlying distributions of features for classes, a task which might be impractical especially when trying to represent concurrent drifts. Secondly, our tool has a broader scope than just generating a dataset. In particular it runs several learners in parallel and in real-time in the experiment and thus enables real-time online-learning showing the effects of changes in the environment so that the experimenter can get a better sense of the immediate consequences of any change in the environment or the test subject's behaviour.

In [9] the authors propose a publicly released *event-based* framework that can be used to generate datasets based on a different categorisation of concept drifts that uses several criteria to divide drifts into mutually exclusive and well-defined categories. The framework works by specifying initial values for parameters and the number/times of drifts with the new parameter values at the corresponding drifts. This avoids the problem of specifying class-conditional pdfs and mixing proportions for each time step, although a smooth gradual drift would still require manually specifying a long list of consecutive time points and parameters. Unlike our tool, where one can specify an arbitrary number of classes, the framework proposed in [9] is constrained to binary class problems. The datasets are generated for four problems: the CIRCLE problem $(x - a)^2 + (y - b)^2 \leq \text{or} > r^2$ where the goal is to recognise points inside/outside the specified circle, the SINE problem $y \leq \text{or} > a \sin(bx + c) + d$ where the goal is to recognise points under or over the specified sine curve, the moving HYPERPLANE problem $y \leq \text{or} > -a_o + \sum_{i=1}^d a_i x_i$ again aiming at identifying points on either side of the hyperplane and the BOOLEAN problem $y = (\text{colour} =_1 \text{or} \neq_1 a \vee_1 \text{or} \wedge_1, \text{shape} =_2 \text{or} \neq_2 b \vee_2 \text{or} \wedge_2, \text{size} =_3 \text{or} \neq_3 c \vee_3 \text{or} \wedge_3$ where the goal is to learn a class description such as $\text{colour} = \text{blue} \vee \text{shape} = \text{rectangle}$. This framework also makes specifying concurrent drifts impractical but we note the authors do take into account customisable noise levels (without specifying the exact nature of them), this contrasts with Narasimhamurthy's framework where noise is left to be encoded in the class-conditional pdfs.

This raises a question about the difference between noise and concept drift, which after all could also be considered as a form of noise. To avoid the problem of modelling noise, noise in the datasets produced by our tool is due to natural variations in the way tasks are performed by humans.

In [7], the authors simulate concept drift within a 3D car driving game based on the XNA Racing Game. The authors try to predict the next driving action of a test subject playing the game and introduce concept drift by changing the driving environment. There are a total of 6 classes (driving actions) that can be predicted. The authors produce two datasets, one with sudden concept drift and one with gradual concept drift. They introduce the drifts by changing the mass of the car which modifies the handling and reactivity for the test subject. Similar to our tool, the authors use an

interactive experiment to generate their data except that they only consider one type of drift at a time. In their experiments, the authors do not consider concurrent drifts and it is unclear how easy it would be to encode these within their framework as it is unreleased. Furthermore, learning on their datasets is done with a sliding window training set and an SVM classifier. We argue that for many human-centric applications such a paradigm is unrealistic and that in real world applications we will have concurrent concept drifts of different types. We anticipate that in such applications gradual and sudden drift will sometimes be present concurrently, moreover for practical reasons, save the initial training phase, it is expected to be difficult to acquire large amounts of labelled data at once (500 instances in their case) for a learner to update itself.

That being said, the data generation part of the framework presented in [7] comes closest to our approach. Its strength comes from the data generation process being interactive with classes and a very concrete environment. Our tool generates data in a more abstract environment but has the advantage of being more flexible and modular. It enables the experimenter to choose the number of features and classes, define and dynamically redefine classes, schedule customisable drifts events and visualise not only the drift happening throughout the interactive data generation process but also the cumulative online prediction performance of any attached learner.

C. Concept drift in body-sensor based activity recognition

In body-sensor based activity recognition, concept drift manifests itself by a change in the distribution of an activity’s features. This can be due to drift in the sensor, a change in the sensor’s physical location due to slippage or simple repositioning after a period of being taken off by a user, or a change in the user’s motor strategy.

The concept drift most researched in activity recognition is a consequence to a change in a sensor’s physical location, referred to as *sensor-displacement*. Currently, there are two main approaches for coping with sensor displacement in body-worn sensor activity recognition. Namely, devising sensor displacement invariant features such that performance is not significantly affected even in the case of sensor displacement and using an unsupervised self calibrated learner.

In [6] the authors use physical properties of sensors to produce a set of heuristics discarding signal frames dominated by rotation which the authors claim are the ones mainly affected by the displacement. In [3] the authors use genetic programming to generate a single invariant feature by composing base features. Finding displacement invariant features is only a partial solution to the concept drift problem in long-term activity recognition, indeed it does not capture the fact that the learner using these features must be able to adapt to non-sensor caused concept drift such as a change in a subject’s motor strategy. Examples of adaptive learner

solutions are given in [2] and [4]. Both approaches are unsupervised and rely on classifier self-calibration. In [2] the authors model sensor displacement by a vector $\vec{\theta}$, they assume that given $\vec{\theta}$ a drifted point with features y can be classified as $C(y - \vec{\theta})$. They keep track of the sensor drift by using an Expectation-Maximisation strategy to estimate $\vec{\theta}$ as time passes. A more flexible approach is given in [4] where the authors use the labels assigned by the learner to update the class centroids in a Nearest Centroid Classifier (NCC). For the self calibration of the NCC, the authors use the update

$$CC_{t+1}^{c_i} = (1 - \alpha)CC_t^{c_i} + \alpha x_t \quad (1)$$

when a sample x_t is classified as belonging to class c_i at time t for c_i ’s class centroid CC^{c_i} , α is the learning rate. Essentially, this approach does not try to suppress concept drift, it aims at keeping track of it by updating its class centroids based on the labels the learner attributes to testing instances. The approach thus does not make any assumption about the drift and for this reason we will also use the NCC classifier as a base for our experiments in the following section. The main limitation of these two approaches is that they are prone to self calibration errors, i.e. using mislabelled data and incorporating it in the model that then strengthens the erroneous classification decision even more. As we later show, this problem, which we call *the mislabeling cycle problem*, is susceptible to appearing in the case of sudden drift and is the main motivation for using semi-supervised learners which have the ability to incorporate ground truth that will enable a supervisor (user) to guide the learner if it starts to diverge. We argue that this is a crucial capability for any system to be useful in practice.

III. INTERACTIVE SIMULATION

Unlike other tools, the concept drift tool that we have developed aims to support the simulation of different types of drift that overlap in time (concurrent drifts), better reflecting how drifts occur in systems with human activities. To capture the natural variations in the way humans perform actions, the data generation process interactively captures and uses human responses to simple point-and-click challenges (e.g. clicking on particular coloured disks in sequence). The tool also allows users to specify the number of features and the classes required, as well as how changes in the environment should be scheduled. Scheduling follows the approach used in [7] which avoids the explicit use of pdfs unlike [10] that requires them. Different (semi/un)supervised online learners can be configured and run simultaneously on the data generating a moving plot of the performance of the learners to drifts as they happen.

Our concept drift simulation tool is shown in Figure 1. The tool produces customisable simulations in which a subject must perform a sequence of (abstract) activities. The

tool is used to test the suitability of learners for applications with realistic constraints. The aim is to understand a learner’s reaction to different drifts including drifts occurring concurrently. The latter case is rarely considered in the literature but we believe it is important and likely to arise in practice. Thus, rather than modelling drift parameters from real data sets which would require us to devise the underlying distribution of features and estimate the types and characteristics of drifts – a task that in practice would either require strong assumptions or be impossible [9], we choose instead to explore parallels between activity recognition and performing an interactive simulation.

An interactive simulation consists of a training phase, which simply asks the subject to complete a certain number of activities and saves the subject’s responses for later training and a continuous learning phase where the user is asked to complete activities while a drift manager applies a schedule of predefined and possibly overlapping concept drifts. All parameters for the simulation and the different learners to be tested are defined in a configuration file.

For our case study, we have produced a simulation, where activities consist of clicking on a sequence of coloured discs, that can be configured to move and/or grow (Figure 1). The features are the x and y positions of the clicks for each disc constituting an activity. We chose to let each person freely choose where to click on discs, although in practice most subjects aim for the centre of discs. Different learners can be tested and their performance is displayed in the lower right graph. The configuration file for the simulation includes the number, colour and initial position of discs, number of discs per activity and number of activities as specified by the experimenter.

In our evaluated simulation we defined a constant (repeated) drift with a small amplitude and a sudden drift set to happen once with large amplitude, the amplitude being the absolute amount of change in a disc’s position. In total the simulation consists of a sequence of 120 equiprobable activities, each taken from a subset of 6 different discs, to be completed by a subject. The following simulation design choices also need to be taken into consideration:

- *natural variability of activities or noise*: are captured by the simulation being interactive and the natural variability of human responses. .
- *sensor caused drift*: in activity recognition, this can be due to sensor slippages or natural sensor-drift. We chose to incorporate this as gradual drift such that each disc centre’s coordinates can change up to 1% on the x -axis and up to 2% on the y -axis randomly at each time step of the interactive experiment.
- *sensor-displacement*: this is usually a relatively infrequent occurrence and is assumed to have arbitrary amplitude, if for example, a sensor is damaged before being put back on. We translate this sudden concept drift as a planned event happening after the first 40

activities are completed by the subject. The sudden drift changes each disc centre’s coordinates by at most 30% both on the x -axis and the y -axis.

- *change in motor strategy*: this is addressed by growing the size of the discs. As discs grow larger and start to overlap, users have to click on different parts of them, rather than close to the centre when the discs are small and non-overlapping. We increase the discs’ radiuses by 3% at each time step.

IV. RESULTS

We now report on the learners that were used for the simulation outlined in Section III and discuss the results obtained pointing to a plausible approach for handling conflict drift in activity recognition.

Similarly to [4] we used the Nearest Centroid Classifier (NCC) as our base classifier, but we also added two modified versions of it to benchmark the effect of adding semi-supervision to the NCC. The first version, called Semi-supervised NCC (SSNCC), implements a function that is called during the simulation providing the learner with a labelled datapoint that the learner uses to perform an update as in Equation 1. This learner thus performs a one step calibration whenever it receives the ground truth. We found empirically that very high learning rates worked best with the SSNCC approach ($\alpha = 0.9$ or 1.0). Intuitively this is not surprising as it amounts to forgetting most of the old centroids and replacing them with the ground truth as it becomes available. If we give labels to the learner often enough this performs well, but as we target real life applications this will be unrealistic. The second modified NCC learner is a Hybrid NCC (HNCC), which perform two types of updates. Self-calibration updates as with the adaptive NCC and semi-supervised one step calibration updates identical to the SSNCC. The HNCC thus has two learning rates, α_1 for self-calibration and α_2 for incorporation of ground truth. Empirically we found that low α_1 ($\simeq 0.15$) and high α_2 (0.9 or 1.0) performed best. Intuitively this seems to be sensible as α_2 weighs how much of the ground truth we decide to incorporate for such an update.

During the simulation the learners were given the label of every 16th activity, this amounts to $\sim 7.5\%$ labelled data, a relatively small amount by most semi-supervised standards, yet arguably still too high for practical purposes.

From our initial set of experiments we observed two main trends: cases similar to Figure 2 where the (unsupervised) adaptive NCC is able to adapt to the concurrent drifts and cases where it is unable to do so as in Figure 3, apparently suffering from cyclic mislabelling such that its performance degrades beyond that of a non-adaptive solution.

In Figure 2 we can observe that the SSNCC learner takes longer than the adaptive NCC and HNCC to restore its performance. This can be explained by the amount of time the learner has to wait to gather enough ground truth to have

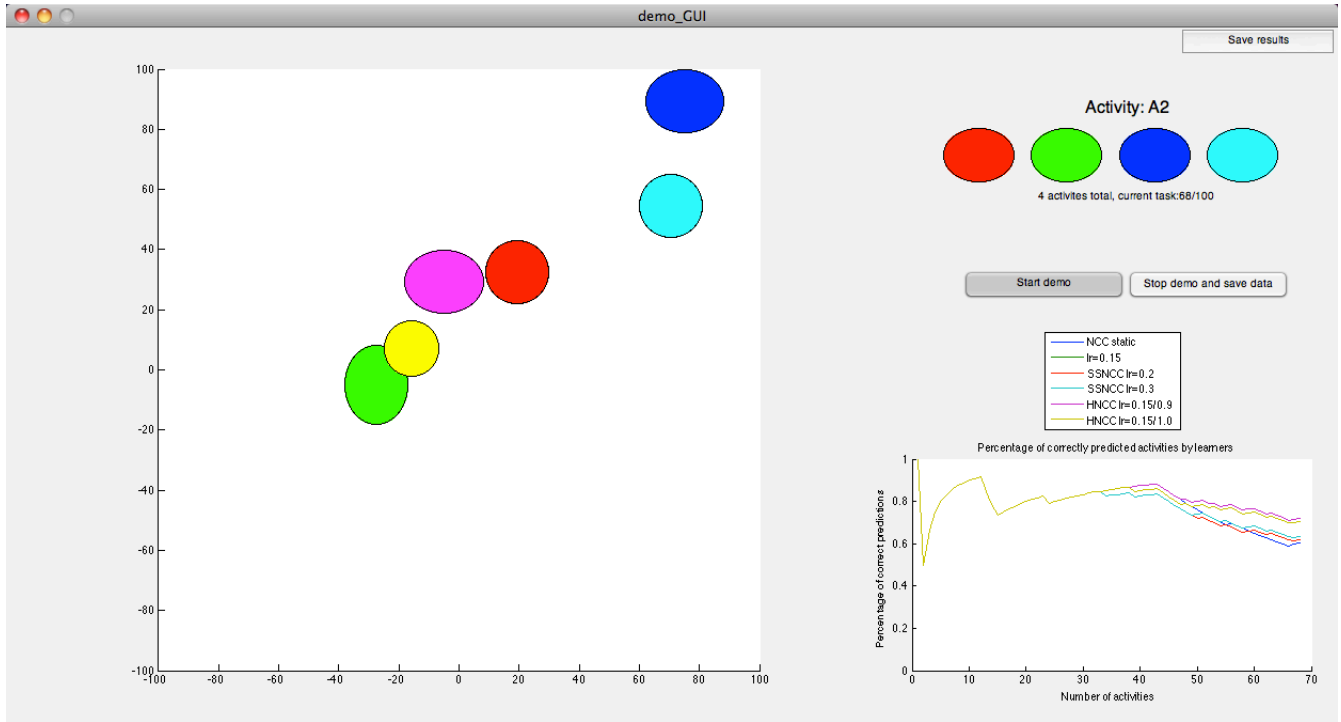


Figure 1: A screen capture of an interactive simulation created using our tool. The user must click the sequences of coloured discs displayed on the top right to complete an activity. The performance of learners is shown in real time on the lower right graph

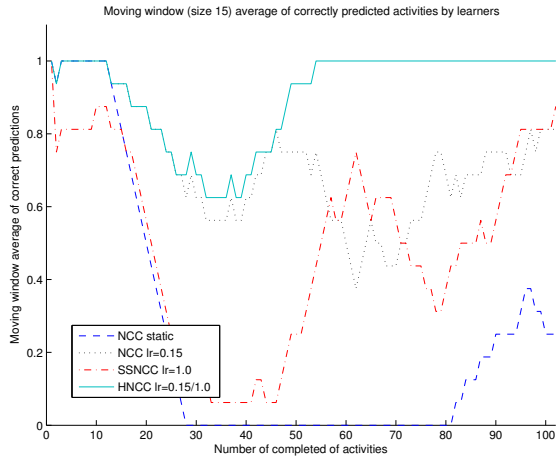


Figure 2: Shows the unsupervised NCC successfully adapting to the concept drift. The semi supervised solution SSNCC manages to restore its performance after being given enough labels while the hybrid HNCC restored performance fastest. lr denotes the learning rate(s) for learners. For the semi-supervised learners labels are given every 16th activity ($\sim 7.5\%$ labelled data).

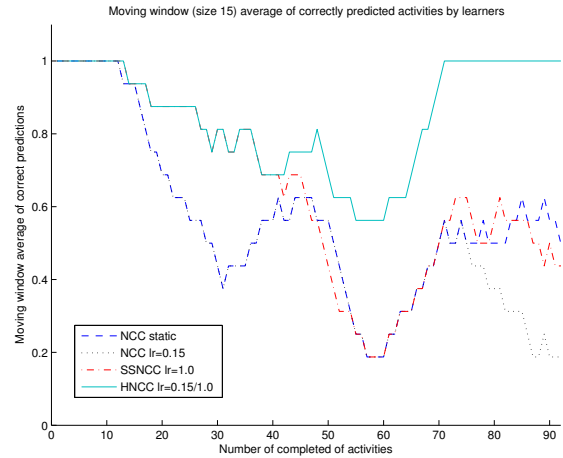


Figure 3: Shows the unsupervised NCC failing to adapt to the concept drift, due to cyclic mislabelling its performance eventually falls below the static NCC solution. The semi supervised solution SSNCC does not manage to restore performance in the time of the experiment given the frequency and nature of provided labels. lr denotes the learning rate, and for the semi-supervised learners labels are given every 16th activity ($\sim 7.5\%$ labelled data).

an updated model of the drifted learning task. The hybrid solution HNCC, on the other hand, benefits from the self-calibration updates of the NCC while taking advantage of infrequent access to ground truth. Empirically, HNCC was

found to be a faster adapting and more robust solution than any of the two learners it is based on.

Conversely, as Figure 3 shows, the NCC learner has not

been able to cope with concurrent drifts and has ended up with a very distorted internal model. This shows the potential risk of unsupervised adaptive learners, and in this particular figure, the drift is such that even the SSNCC learner does not receive enough labels to restore performance. The higher accuracy and faster adaption of the HNCC solution can be explained by the self-calibration which is been done based on ground truth. This enables the HNCC to avoid the cyclic mislabelling problem, as the cycles will be broken as soon as the HNCC learner receives a label for the mislabelled classes given a high enough α_2 learning rate.

V. DISCUSSION

We have constructed a tool enabling the creation of interactive simulations with customisable concept drift events. These are used to test the suitability of learners when faced with *single or concurrent* concept drift occurring in models of human-centric tasks like activity recognition. Our initial work suggests that hybrid, self-calibrating learners that are also able to incorporate ground truth might be appropriate for robust body-worn sensor activity recognition carried out over extended periods of time where concurrent concept drifts are likely to occur. This hypothesis is also supported by other work in long-term activity recognition [12] that investigated techniques to lower the number of labels required by fully supervised approaches.

We also identified and highlighted the problem of cyclic mislabelling in state-of-the-art (unsupervised) self-calibrating solutions for activity recognition. The risk with fully unsupervised learning is that a learner might reinforce its misclassification errors and thus might use erroneous labels for self learning. We expect that any learner will make classification errors if deployed over long enough periods of time and thus argue that a fully unsupervised solution, although ideal, is not feasible in practice. Our interactive simulations show that a semi-supervised learner has the potential to recover performance even after a severe alteration in the distribution of the input data as it gets increasing amounts of ground truth. Finally, we have seen that a hybrid approach, combining frequent unsupervised self-calibration and infrequent semi-supervised learning, can achieve the minimum performance of a simple semi-supervised learner while taking advantage of self calibration to increase adaption speed and robustness.

Future work will focus on validating our hypothesis that a hybrid approach is more robust than a semi-supervised or unsupervised adaptive classifier on real datasets, starting with the recent activity recognition benchmark [1] that is concerned with sensor-displacement.

VI. ACKNOWLEDGEMENTS

This research was supported by EU FP7 research grant 264738 (iCareNet).

REFERENCES

- [1] O. Baños, M. Damas, H. Pomares, I. Rojas, M. Tóth, and O. Amft. A benchmark dataset to evaluate sensor displacement in activity recognition. In *UbiComp '12*, 2012.
- [2] J. Bayati, H. and del R. Millán and R. Chavarriaga. Un-supervised adaptation to on-body sensor displacement in acceleration-based activity recognition. In *ISWC '11*, 2011.
- [3] K. Förster, P. Brem, D. Roggen, and G. Tröster. Evolving discriminative features robust to sensor displacement for activity recognition in body area sensor networks. In *ISSNIP 2009*, 2009.
- [4] K. Forster, D. Roggen, and G. Troster. Unsupervised classifier self-calibration through repeated context occurrences: Is there robustness against sensor displacement to gain? In *ISWC '09*, 2009.
- [5] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I. Vlahavas. An adaptive personalized news dissemination system. *J. Intell. Inf. Syst.*, 2009.
- [6] K. Kunze and P. Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *UbiComp '08*, 2008.
- [7] P. Lindstrom, S. J. Delany, and B. MacNamee. Autopilot: Simulating changing concepts in real data. In D. Bridge, K. Brown, B O'Sullivan, and H Sorensen, editors, *AICS-08*, 2008.
- [8] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *UbiComp '07*, 2007.
- [9] L. Minku, A. White, and X. Yao. The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Trans. on Knowl. and Data Eng.*, 2010.
- [10] A. Narasimhamurthy and L. Kuncheva. A framework for generating data to simulate changing environments. In *AIAP'07*, 2007.
- [11] M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.
- [12] M. Stikic, K. Van Laerhoven, and B. Schiele. Exploring semi-supervised and active learning for activity recognition. In *ISWC '11*, 2008.
- [13] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen. Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In *CBMS*, 2006.
- [14] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 1996.
- [15] I. Zliobaite. Learning under concept drift: an overview. *CoRR*, abs/1010.4784, 2010.