Unsupervised Concept Drift Detection using a Student–Teacher Approach

Vitor Cerqueira[1]✉, Heitor Murilo Gomes[2], and Albert Bifet[2,3]

[1] LIAAD-INESCTEC, Porto, Portugal
[2] University of Waikato, Hamilton, New Zealand
[3] Télécom ParisTech, Paris, France
cerqueira.vitormanuel@gmail.com

Abstract. Concept drift detection is a crucial task in data stream evolving environments. Most of the state of the art approaches designed to tackle this problem monitor the loss of predictive models. Accordingly, an alarm is launched when the loss increases significantly, which triggers some adaptation mechanism (e.g. retrain the model). However, this *modus operandi* falls short in many real-world scenarios, where the true labels are not readily available to compute the loss. These often take up to several weeks to be available. In this context, there is increasing attention to approaches that perform concept drift detection in an unsupervised manner, i.e., without access to the true labels. We propose a novel approach to unsupervised concept drift detection, which is based on a student-teacher learning paradigm. Essentially, we create an auxiliary model (student) to mimic the behaviour of the main model (teacher). At run-time, our approach is to use the teacher for predicting new instances and monitoring the *mimicking* loss of the student for concept drift detection. In a set of controlled experiments, we discovered that the proposed approach detects concept drift effectively. Relative to the gold standard, in which the labels are immediately available after prediction, our approach is more conservative: it signals less false alarms, but it requires more time to detect changes. We also show the competitiveness of our approach relative to other unsupervised methods.

Keywords: Concept drift detection · Data streams · Unsupervised learning · Model compression

1 Introduction

Learning from data streams is a continuous process. When predictive models are deployed in environments susceptible to changes, they must detect these changes and adapt themselves accordingly. The phenomenon in which the data distribution evolves is referred to as *concept drift*, and a sizeable amount of literature has been devoted to it [7].

Concept drift detection and adaptation are typically achieved by coupling predictive models with a change detection mechanism [10]. The detection algorithm launches an alarm when it identifies a change in the data. Typical concept

drift strategies are based on sequential analysis [17], statistical process control [6], or monitoring of distributions [3]. When change is detected, the predictive model adapts by updating its knowledge with recent information. A simple example of an adaptation mechanism is to discard the current model and train a new one from scratch. Incremental approaches are also widely used [9].

The input data for the majority of the existing drift detection algorithms is the performance of the predictive model over time, such as the error rate. In many of these detection methods, alarms are signalled if the performance decreases significantly. However, in several real-world scenarios, labels are not readily available to estimate the performance of models. Some labels might arrive with a delay or not arrive at all due to labelling costs. This is a major challenge for learning algorithms that rely on concept drift detection as the unavailability of the labels precludes their application [10].

In this context, there is increasing attention toward unsupervised approaches to concept drift detection. These assume that, after an initial fit of the model, no further labels are available during the deployment of this model in a test set. Most works in the literature handle this problem using statistical hypothesis tests, such as the Kolmogorov-Smirnov test. These tests are applied to the output of the models, either the final decision or the predicted probability.

1.1 Contributions and Paper Organisation

Our goal in this paper is to address concept drift detection in an unsupervised manner. To accomplish this, we propose a novel approach to tackle this problem using a student-teacher (ST) learning paradigm. The gist of the idea is as follows. On top of the main predictive model, which we designate as the teacher, we also build a second predictive model, the student. Following the literature on model compression [5] and knowledge distillation [13], the student model is designed to mimic the behaviour of the teacher.

Using the ST framework, our approach to unsupervised concept drift detection is carried out by monitoring the mimicking loss of the student. The mimicking loss is a function of the discrepancy between the prediction of the teacher and the prediction of the student in the same instance. In summary, we use the loss of the student model as a surrogate for the behaviour of the main model. Accordingly, we can apply any state of the art approach in the literature which takes the loss of a model as the main input, for example, ADWIN [3] or the Page-Hinkley test [17].

When concept drift occurs, it causes changes in the prior probabilities of the classes or changes in the class conditional probabilities of the predictor variables. In effect, we hypothesise that these changes disrupt the collective behaviour between the teacher and student models. In turn, this change of behaviour may be captured by monitoring the mimicking loss of the student model.

We validate the proposed method using a set of experiments with an artificial drift process, which we adapt from Žliobaite [22]. The proposed method is

publicly available online to support reproducible science[4]. Our implementation is written in Python and is based on the scikit-multiflow framework [16].

2 Background

2.1 Problem Definition

Let $D(X, y) = \{(X_1, y_1), \ldots, (X_t, y_t)\}$ denote a data stream, where each X is a q-dimensional array representing the input predictor variables. Each y represents the corresponding output label. We assume that the values of y are categorical. The goal is to use this data set $\{X_i, y_i\}_1^t$ to create a classification model to approximate the function which maps the input X to the output y. Let $\mathcal{T}$ denote this classifier. The classifier $\mathcal{T}$ can be used to predict the labels of new observations X. We denote the prediction made by the classifier as $\hat{y}_{\mathcal{T}}$.

Many real-world scenarios exhibit a non-stationary nature. Often, the underlying process causing the observations changes in an unpredictable way, which degrades the performance of the classifier $\mathcal{T}$. Let $p(X, y)$ denote the joint distribution of the predictor variables X and the target variable y. According to Gama et al. [7], concept drift occurs if $p(X, y)$ is different in two distinct points in time across the data stream. Changes in the joint probability can be caused by changes in $p(X)$, the distribution of the predictor variables or changes in the class conditional probabilities $p(X|y)$ [8]. These may eventually affect the posterior probabilities of classes $p(y|X)$.

2.2 Label Availability

When concept drift occurs, the changes need to be captured as soon as possible, so the decision rules of $\mathcal{T}$ can be updated. The vast majority of concept drift detection approaches in the literature focus on tracking the predictive performance of the model. If the performance degrades significantly, an alarm is launched and the learning system adapts to these changes.

The problem with these approaches is that they assume that the true labels are readily available after prediction. In reality, this is rarely the case. In many real-world scenarios, labels can take too long to be available, if ever. If labels do eventually become available, often we only have access to a part of them. This is due to, for example, labelling costs. The different potential scenarios when running a predictive model are depicted in Figure 1.

Precisely, a predictive model is built using an initial batch of training data, whose labels are available. When this model is deployed in a test set, concept drift detection is carried out in an unsupervised or supervised manner.

In unsupervised scenarios, no further labels are available to the predictive model. Therefore, concept drift detection must be carried out using a different strategy other than monitoring the loss. For example, one can track the output probability of the models [22].

[4] `https://github.com/vcerqueira/unsupervised_concept_drift`

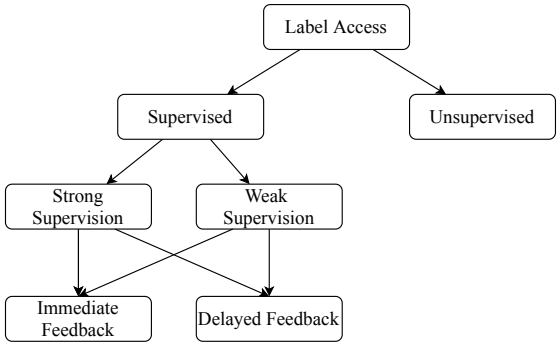

Fig. 1. The distinct potential scenarios regarding label access after the initial fit of the model (adapted from Gomes et al. [9]).

Concept drift detectors have access to labels when the scenario is supervised. On the one hand, the setting may be either strongly supervised or weakly supervised [21]. In the former, all labels become available. In the latter, the learning system only has access to a part of the labels. This is common in applications which data labelling is costly. On the other hand, labels can arrive immediately after prediction, or they can arrive with some delay. In some domains, this delay may be too large, and unsupervised approaches need to be adopted.

In this paper, we address concept drift detection from an unsupervised perspective. In this setting, we are restricted to use $p(X)$ to detect changes, as the probability of the predictor variables is not conditioned on y.

3 Related Research

3.1 Concept Drift Detection

Concept drift can occur in three different manners: suddenly, in which the current concept is abruptly replaced by a new one; gradually, when the current concept slowly fades; and reoccurring, in which different concepts are prevalent in distinct time intervals (for example, due to seasonality).

We split concept drift detection into two dimensions: supervised and unsupervised. The supervised type of approaches assumes that the true labels of observations are available after prediction. Hence, they use the error of the model as the main input to their detection mechanism. On the other hand, unsupervised approaches preclude the use of the labels in their techniques.

Plenty of error-based approaches have been developed for concept drift detection. These usually follow one of three sort of strategies: sequential analysis, such as the Page-Hinkley test (PHT) [17]; statistical process control, for example the Drift Detection Method (DDM) [6] or the Early Drift Detection Method (EDDM) [1]; and distribution monitoring, for example the Adaptive Windowing (ADWIN) approach [3].

Although the literature is scarce, there is an increasing interest in approaches which try to detect drift without access to the true labels. Žliobaite [22] presents a pioneer work of this type. She proposed the application of statistical hypothesis testing to the output of the classifier (either the probabilities or the final categorical decision). The idea is to monitor two samples of one of these signals. One sample serves as the reference window, while the other represents the detection window. When there is a statistical difference between these, an alarm is launched. In a set of experiments, Žliobaite shows that concept drift is detectable using this framework. The hypothesis tests used in the experiments are the two-sample Kolmogorov-Smirnov test, the Wilcoxon rank-sum test, and the two-sample t-test.

Reis et al. [19] follow a strategy similar to Žliobaite [22]. They propose an incremental version of the Kolmogorov-Smirnov test and use this method to detect changes. In the same line of research, Yu et al. [20] apply two layers of hypothesis testing hierarchically. Kim et al. [14] also apply a windowing approach. Rather than monitoring the output probability of the classifier, they use a confidence measure as the input to drift detectors.

Pinto et al. [18] present an automatic framework for monitoring the performance of predictive models. Similarly to the above-mentioned works, they perform concept drift detection based on a windowing approach. The signal used to detect drift is computed according to a mutual information metric, namely the Jensen-Shannon Divergence [15]. The window sizes and threshold above which an alarm is launched is analysed, and the approach is validated in real-world data sets. The interesting part of the approach by Pinto et al. [18] is that their method explains the alarms. This explanation is based on an auxiliary binary classification model. The goal of applying this model is to rank the events that occurred in the detection window according to how these relate to the alarm. These explanations may be crucial in sensitive applications which require transparent models.

Gözüaçık et al. [11] also develop an auxiliary predictive model for unsupervised concept drift detection, which is called D3 (for (Discriminative Drift Detector). The difference to the work by Pinto et al. [18] is that they use this model for detecting concept drift rather than explaining the alarms.

3.2 Student–Teacher Learning Approach

Model compression, also known as student-teacher (ST) learning, is a technique presented by Buciluǎ et al. [5]. The goal is to train a model, designated as a student, to mimic the behaviour of a second model (the teacher). The authors use this approach to compress a large ensemble (the teacher) into a compact predictive model (the student).

Hinton et al. [13] developed the idea of model compression further, denoting their compression technique as knowledge distillation. Distillation works by softening the probability distribution over classes in the softmax output layer of a neural network. The authors address an automatic speech recognition problem

by distilling an ensemble of deep neural networks into a single and smaller deep neural network.

Both Buciluǎ et al. [5] and Hinton et al. [13], show that combining the predictions of the ensemble leads to a comparable performance relative to a single compressed model.

While our concerns are not about decreasing the computational costs of a model, we can leverage model compression approaches to tackle the problem of concept drift detection. Particularly, by creating a student model which mimics the behaviour of a classifier, we can perform concept drift detection using the loss of the student model. Since this loss is not conditioned on the target variable y, concept drift detection is carried out in an unsupervised manner.

4 Methodology

In this section, we formalise our approach to concept drift detection. Our method is based on a student-teacher (ST) learning approach. The only information required from the environment is predictor variables of testing instances (X). Since the proposed method is not conditioned on the labels of the target variable (y), we refer to it as unsupervised.

From a high-level perspective, the proposed approach settles on three main steps:

1. Creating the main model $\mathcal{T}$, which is the teacher;
2. Creating the student model $\mathcal{S}$, which mimics the behaviour of $\mathcal{T}$;
3. Deploying the main model $\mathcal{T}$ and performing concept drift detection based on the loss of $\mathcal{S}$;

In the next subsections, we will detail each step in turn.

4.1 Creating the Teacher and Student Models

Main Classifier $\mathcal{T}$ Let $D_{tr}(X, y)$ denote the available training instances. We use $D_{tr}(X, y)$ to train the classifier $\mathcal{T}$, where $D_{tr}(X, y)$ is an initial batch of training instances. This model is used to make predictions on new upcoming instances in the stream D, which we denote as X_{new}. We assume that the model is incremental [9]. $\mathcal{T}$ is updated when new labels become available.

Student–Teacher Approach We assume that the corresponding labels of X_{new} are not available for a long period after making the prediction. Hence, we cannot rely on approaches that monitor the loss of $\mathcal{T}$ to detect concept drift.

We adopt a student-teacher (ST) learning approach to circumvent this problem. In the ST framework, $\mathcal{T}$ is the teacher model. Then, we create a second predictive model $\mathcal{S}$, the student, which is trained to mimic the behaviour of the teacher, $\mathcal{T}$. This is accomplished as follows. We obtain the predictions $\hat{y}_{\{\mathcal{T},tr\}}$ of $\mathcal{T}$ in the available data D_{tr} used to create it. In effect, we can set up a new

data set $D_{tr}(x, \hat{y}_{\mathcal{T}})$, which can be used to train $\mathcal{S}$. In other words, we train the student model using the same observations used to train the teacher. However, the target variable is replaced with the predictions of the teacher.

It might be argued that using the same instances to train both the teacher and the student models leads to overfitting. However, Hinton et al. [13] show that this is not a concern.

In the typical student-teacher approaches, designed for model compression [5] or knowledge distillation [13], the goal is to compress a model with a large number of parameters (usually an ensemble) into a more compact model with comparable predictive performance. In these cases, the student model is deployed in the test set, while the teacher is not used. Conversely, in our methodology, we leverage the student-teacher framework differently. The student model is regarded as a model which can make predictions regarding the behaviour of $\mathcal{T}$, i.e., what the output of $\mathcal{T}$ will be for a given input observation. Both $\mathcal{T}$ and $\mathcal{S}$ models are used in practice, as explained below.

4.2 Concept Drift Detection

Since we assume that the true labels are unavailable, we cannot measure the loss of the main model, $\mathcal{T}$. But we can measure the loss of the student model: the discrepancy between the prediction of $\mathcal{T}$ ($\hat{y}_{\mathcal{T}}$) and the prediction of $\mathcal{S}$ about $\hat{y}_{\mathcal{T}}$ ($\hat{y}_{\mathcal{S}}$). The loss of $\mathcal{S}$ is then defined as $L(\hat{y}_{\mathcal{T}}, \hat{y}_{\mathcal{S}})$, where L is the loss function, for example, the error rate.

In effect, our approach to unsupervised concept drift detection is to monitor the error of the student model. This can be accomplished with any state of the art concept drift detection approach, e.g. ADWIN [3]. Essentially, we use the loss of the student model as a surrogate signal for concept drift detection. While $\mathcal{S}$ is monitored for detecting drift, the model $\mathcal{T}$ is used to make predictions on new instances X_{new}.

Our working hypothesis is the following. When concept drift occurs, it potentially causes changes in the posterior probability of classes $p(y|X)$. Consequently, this change in the behaviour of $\mathcal{T}$ will perturb the mimicking loss of the student model, $L(\hat{y}_{\mathcal{T}}, \hat{y}_{\mathcal{S}})$. Therefore, tracking this signal may enable us to capture changes in the environment without access to any labels.

5 Experiments

In this section, we detail the experiments carried out to validate the proposed approach to unsupervised concept drift detection.

The experiments are designed to address the following research questions:

- **RQ1**: Is the proposed unsupervised ST approach able to detect concept drift in the data?
- **RQ2**: How does the proposed method compare with the gold standard, in which all the labels are immediately available after prediction? Note that

even though this scenario is unlikely in real applications, it may still serve as a benchmark of performance for other approaches;
- **RQ3**: How does the proposed approach compare with other unsupervised approaches, namely the statistical hypotheses tests described by Žliobaite [22]?
- **RQ4**: Finally, what is the relative drift detection performance between the different label availability scenarios (see Figure 1)?

We used two data sets in our experiments: electricity demand [12], and forest cover type [2]. The electricity data set refers to the electricity market in Australian New South Wales. There are a total of 45.312 observations in the data set, which are captured every half-hour. There are eight predictor variables, all of which numeric. The predictive task is binary classification; to predict whether the price will go up or down relative to a moving average of 24 hours.

The second data set represents the forest cover type and was obtained by the US Forest Service. In the data set, there is a total of 581.012 observations and 54 predictor variables (10 of which numeric, and the remaining are binary). The data set contains five classes regarding the cover type. Both these data sets have been used in multiple works on data stream mining [6, 9, 2].

5.1 Synthetic Drift Injection

We perform experiments using artificial drift in order to understand better the relative behaviour of drift detectors in different scenarios. This is accomplished by following a process similar to that described by Žliobaite [22]. We assume that the initial 60% of the observations are labelled and that these observations are used for an initial fit of the models $\mathcal{T}$ and $\mathcal{S}$. Then, we proceed as follows.

1. We randomly select a point between 70% and 90% of the total observations available. After this point, all subsequent observations are *contaminated* with drift (see Figure 2);
   - Note that we leave a 10% interval on each side (after 60% and before 100%) for securing enough observations to evaluate the behaviour of a concept drift detector; for example, its rate of false alarms or its reactiveness to drift;
2. We randomly select half of the predictor variables from a randomly selected class. The values of these variables are randomly shuffled. Žliobaite [22] handpicks the columns to be swapped. Conversely, we introduce randomness. Essentially, this process injects drift in the conditional probabilities $p(X|y)$.

In order to produce a robust estimate of performance, this process is repeated 50 times in a Monte Carlo approximation manner.

5.2 Methods

We carry out a learning plus testing cycle for each one of the 50 Monte Carlo repetitions. We use an Adaptive Random Forest (ARF) as learning algorithm [9]

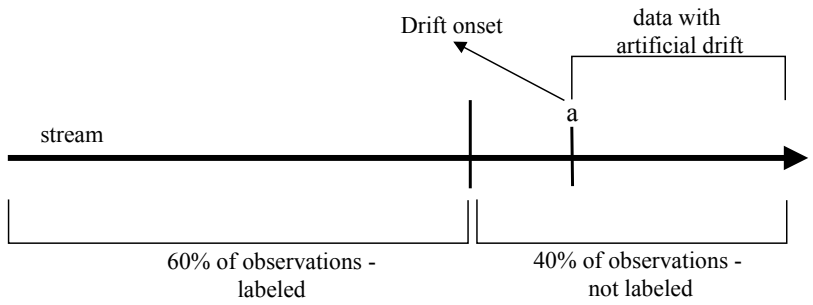

Fig. 2. Workflow for injecting drift. This random process is repeated 50 times following a Monte Carlo approximation approach.

for training both $\mathcal{T}$ and $\mathcal{S}$. As the name implies, the ARF method extends the widely used Random Forest approach by Breiman [4] to evolving data stream classification problems.

We focus on performing concept drift detection in an unsupervised manner. Notwithstanding, we also test several variants of supervised scenarios. All of these are detailed below.

Unsupervised Approaches (U) After the initial training with 60% of the observations, our unsupervised setup assumes that no further label is available. For each upcoming instance, we only have access to the predictor variables (X). Accordingly, as we described in Section 4, concept drift is performed by monitoring the error of $\mathcal{S}$. In this scenario, we also apply the ADWIN and PHT methods using the error rate of $\mathcal{S}$. These are denoted as `U-ADWIN` and `U-PHT`, respectively. Note that the model $\mathcal{S}$ can be updated online, because its labels are the predictions of model $\mathcal{T}$.

As benchmarks, we also include the following statistical tests suggested by Žliobaite [22]:

- `U-KS`: The two-sample Kolmogorov-Smirnov test, which tests whether two samples come from the same distribution;
- `U-WRS`: The Wilcoxon rank sum test, which tests whether two samples have equal medians;
- `U-TT`: The two-sample t-test, which tests whether two samples have equal means;

Each of these tests are applied using the class output predicted by $\mathcal{T}$ using a sliding window fashion [22]. Specifically, suppose we are at time step i. We create two contiguous samples of the same size (w) up to point i (see Figure 3). The first sample, which represents the reference window, includes the data in the interval $[i-2\times w+1; i-w]$. The second sample, which denotes the detection window [22], contains the information in the interval $[i-w+1; i]$. An hypothesis

test is carried out using these two samples. An alarm is issued, and $i + 1$ is a change point, if the p-value returned by the test is below α.

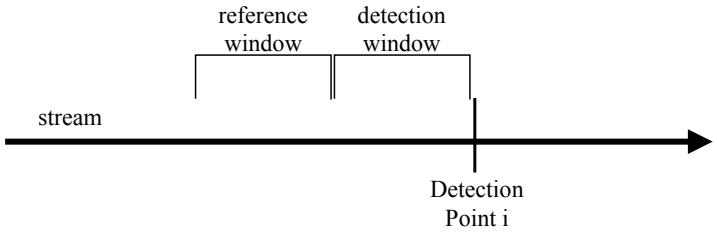

Fig. 3. Detection framework based on windowing.

Supervised Approaches We also include the following supervised approaches in our experiments.

- **Strongly Supervised** (**SS**): We apply the typical procedure which assumes that all the true labels are immediately available after making a prediction. This can be regarded as the gold standard. The term *strong* means refers to the fact that all labels are available during testing [21];
- **Weakly Supervised** (**WS**): In many real-world scenarios, particularly in high-frequency data streams, data labelling is costly. Hence, predictive models can only be updated using a part of the entire data set. This process is commonly referred to as weakly supervised learning [21]. We simulate a weakly supervised scenario in our experiments. Accordingly, predictive models only have access to *l_access*% of the labels. In other words, after a model predicts the label of a given instance, the respective label is immediately available with a *l_access*% chance;
- **Delayed Strongly Supervised** (**DSS**): Labels can take some time to arrive. We study this aspect by artificially delaying the arrival of the labels by *l_delay* instances. After a label becomes available, the respective predictive model is updated;
- **Delayed Weakly Supervised** (**DWS**): We combine the two previous scenarios. In the delayed weakly supervised setup, only *l_access*% of the labels are available. Those which are available arrive with a delay of *l_delay* observations.

In the supervised variants, the classifier $\mathcal{T}$ is updated online as soon as each new label is available. In all these variants, concept drift detection is carried out using the error rate of $\mathcal{T}$ with two state of the art approaches: the ADWIN [3] method and the PHT [17] approach. We denote these approaches as *Prefix*-ADWIN and *Prefix*-PHT, respectively. For example, SS-ADWIN refers to the ADWIN method applied using a strongly supervised model.

Parameter Setup The significance level for the hypothesis tests, also known as p-value, is set to 0.001. The window size for carrying these tests is set to 500 for the electricity data set and 1500 for the cover type data set (see Figure 3). As Žliobaite [22] points out, these values usually depend on the task at hand. The number of trees in the ARF models is set to 25. ARF comprises an internal concept drift detection mechanism based on ADWIN [3]. Since our goal is to detect drift, we disabled this process within the ARF model. We set *l_access* to 50 for both data sets. This setup means that, in the weakly supervised schemes, only about 50% of observations are available. The value of *l_delay* was set to 1000 on the electricity data set, and 5000 on the cover type data set. These values were set arbitrarily; the difference between data sets is related to their difference is the sample size. The remaining parameters are set to default according to their respective implementation.

5.3 Evaluation Metrics

We apply the following metrics described by Bifet [2] to evaluate the drift detectors:

- Mean Time between False Alarms (MTFA): How often there is a false alarm when there is no change (the higher, the better). MTFA is measured by averaging the distance between consecutive (false) alarms before the change point. Moreover, this score is also averaged across the 50 Monte Carlo repetitions;
- Mean Time to Detection (MTD): After a change occurs, how long it takes for the method to detect it, on average (the lower, the better). In practice, we measure the number of points between the change point and the next alarm launched by the respective method. Similarly to MTFA, the score of MTD is averaged across the 50 repetitions;
- Missed Detection Ratio (MDR): The probability of failing to detect a drift. This is measured by taking the fraction of repetitions (across the 50 simulations) in which the drift method fails to launch an alarm after the onset of the drift. Ideally, this value should be zero, meaning all drifts are captured irrespective of how long it takes to accomplish this;

On top of these, we also include the total number of detections (ND) launched by a model. This metric is also averaged across the 50 repetitions.

5.4 Results

The results of our experiments are reported in Tables 1 and 2, for the electricity and cover type data sets, respectively. For the MTFA, MTD, and ND metrics, we also include the standard deviation of the results across the 50 Monte Carlo repetitions.

The first research question is related to the analysis of the ability of the proposed approach to detect concept drift (RQ1). According to the results obtained, the proposed methods (`U-ADWIN`, `U-PHT`) have an MDR of 12 and 6%

Table 1. Results on the electricity dataset

Method	MTFA	MTD	MDR	ND
SS-ADWIN	$18\,372 \pm 29\,514$	986 ± 689	0.00	7 ± 2
SS-PHT	1024 ± 284	508 ± 687	0.88	9 ± 4
DSS-ADWIN	677 ± 216	525 ± 572	0.00	19 ± 4
DSS-PHT	357 ± 60	346 ± 290	0.00	26 ± 5
WS-ADWIN	$21\,287 \pm 34\,634$	1229 ± 646	0.00	6 ± 2
WS-PHT	1355 ± 451	503 ± 301	0.90	8 ± 5
DWS-ADWIN	6749 ± 1672	1274 ± 1448	0.00	10 ± 3
DWS-PHT	517 ± 158	533 ± 629	0.06	21 ± 7
U-ADWIN	$11\,439 \pm 30\,002$	2786 ± 3225	0.12	4 ± 4
U-PHT	$11\,823 \pm 28\,244$	3050 ± 4778	0.18	4 ± 2
U-WRS	2040 ± 158	597 ± 246	0.04	17 ± 3
U-TT	611 ± 36	316 ± 284	0.00	27 ± 2
U-KS	2071 ± 200	574 ± 268	0.04	16 ± 3

(U-ADWIN) and 18 and 17% (U-PHT) in the electricity and cover type data sets, respectively. This result shows that most of the drifts introduced synthetically were captured by both approaches. Moreover, it also shows that the proposed concept drift detection methodology is not constrained to a single detector: both ADWIN and PHT present a good detection ability. Other approaches could be used for detection, e.g. DDM [6] or EDDM [1]. We focus on only two methods for two main reasons: first, these are representative of the state of the art; second, the detection method is orthogonal to our contributions. Therefore, we designed the experiments to show the effectiveness of the proposed method rather than comparing the performance of multiple detectors.

Relative to the gold standard (RQ2), the proposed approach is more conservative. Both U-ADWIN, and U-PHT present a higher MTFA and MTD scores relative to a strongly supervised approach in most of the cases. The unsupervised approaches take more time to detect changes (higher MTD). However, they also show a larger interval between false alarms (higher MTFA) – except for the SS-ADWIN approach.

We also compare the hypothesis tests suggested by Žliobaite [22] with the proposed approach (RQ3). Overall, the statistical tests are more sensitive as they launch more alarms (column ND). The MDR is close or equal to zero in all variants, which means they capture almost all the drifts injected in the data.

Finally, we analyse the different scenarios in terms of label availability (RQ4). We start by comparing the strongly supervised scenario with their weakly supervised counterpart, i.e., SS with WS and DSS with DWS. Overall, the weakly supervised approaches tend to launch fewer alarms. The results suggest that having less information leads to more conservative behaviour by the ADWIN and PHT detectors.

We now analyse the impact of delaying information. This is done by comparing the SS variants with the DSS variants, and the WS variants with DWS

Table 2. Results on the cover type dataset

Method	MTFA	MTD	MDR	ND
SS_ADWIN	5806 ± 3354	3070 ± 2919	0.00	12 ± 2
SS_PHT	6317 ± 1117	6012 ± 8017	0.22	14 ± 6
DSS_ADWIN	1532 ± 493	2154 ± 2802	0.00	28 ± 4
DSS_PHT	389 ± 44	406 ± 410	0.00	140 ± 34
WS_ADWIN	$14\,731 \pm 6036$	4185 ± 3579	0.00	9 ± 2
WS_PHT	7099 ± 2286	$10\,200 \pm 13\,882$	0.28	16 ± 7
DWS_ADWIN	4925 ± 1117	2689 ± 3043	0.00	14 ± 3
DWS_PHT	629 ± 477	5441 ± 5902	0.06	42 ± 16
U_ADWIN	$15\,792 \pm 3213$	5411 ± 6465	0.06	10 ± 4
U_PHT	$21\,864 \pm 7720$	5877 ± 6857	0.17	6 ± 3
U_WRS	2055 ± 348	1341 ± 1118	0.00	25 ± 2
U_TT	2208 ± 384	1416 ± 1056	0.00	24 ± 3
U_KS	1771 ± 138	850 ± 867	0.00	30 ± 2

variants. Contrary to weak supervision, delaying the arrival of the labels appears to lead to much more sensitive detectors. Note that these results are constrained on many aspects, for example, data sets, parameter setup, or the drift synthetic process.

5.5 Discussion

In the experiments, we showed that the proposed student-teacher approach can detect concept drift. While it shows a more conservative behaviour relative to the gold standard, the probability of detecting a drift is comparable.

We focused on two state of the art drift detection approaches; ADWIN and PHT. The underlying method applied is orthogonal to our contributions, and we designed the experiments to show the usefulness of the student-teacher approach to unsupervised concept drift detection. In this context, other detectors can be used, such as DDM [6] or EDDM [1].

We controlled the experiments by injecting artificial concept drift in the conditional probabilities $p(X|y)$. This was achieved by randomly swapping the predictors variables in a randomly selected class. The goal of this synthetic process was to enable us to analyse better how the different detection approaches react to drift. In future work, we will develop this analysis from two perspectives:

1. We can analyse the behaviour of the detectors in the presence of drift in the class priors, $p(y)$. To accomplish this, we can follow the strategy by Žliobaite [22], which deletes randomly selected instances from a selected class;
2. We will study the application of the proposed method in a real-world setup without any synthetic process. For example, we can measure its impact by computing the difference in predictive performance. Alternatively, a trade-off between predictive performance and the cost of retrieving a batch on labels to run a supervised approach.

Besides showing the usefulness of the proposed method, we also analysed the behaviour of the two detectors (ADWIN and PHT) under different supervised conditions. Specifically, whether the supervision was strong or weak, in which the latter means that only part of the labels become available. We also analysed the impact of feedback delay, in which the labels take a fixed time to arrive.

In future work, we will extend this analysis. For example, we will perform a sensibility analysis to study how these conditions affect not only the performance of drift detectors but also the performance of the predictive models. We will also evaluate the proposed approach on purely real data sets without any artificial mechanisms.

6 Final Remarks

The literature for concept drift detection is mostly focused on detecting changes by discovering significant deviations in the loss of the model. In this paper, we follow the hypothesis that it is too optimistic to assume that the labels are readily available for computing the loss [22, 18]. Therefore, we tackle the concept drift detection problem in an unsupervised manner.

We develop a novel approach based on an ST learning paradigm. ST approaches are commonly applied to model compression [5] or knowledge distillation [13]. To our knowledge, this is the first work attempting to use an ST approach for concept drift detection.

We validate our proposal with synthetic experiments using two benchmark data sets. The results are promising. The developed method can detect the drifts induced artificially as well as the gold standard, which represents the approach that assumes that labels are immediately available after prediction. Our approach is more conservative relative to the gold standard, and competitive relative to other unsupervised baseline approaches.

Acknowledgements

This project was financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

References

1. Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R.: Early drift detection method. In: Fourth international workshop on knowledge discovery from data streams. vol. 6, pp. 77–86 (2006)
2. Bifet, A.: Classifier concept drift detection and the illusion of progress. In: International Conference on Artificial Intelligence and Soft Computing. pp. 715–725. Springer (2017)
3. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. pp. 443–448. SIAM (2007)

4. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
5. Buciluǎ, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 535–541. ACM (2006)
6. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Brazilian symposium on artificial intelligence. pp. 286–295. Springer (2004)
7. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
8. Gao, J., Fan, W., Han, J., Yu, P.S.: A general framework for mining concept-drifting data streams with skewed distributions. In: Proceedings of the 2007 siam international conference on data mining. pp. 3–14. SIAM (2007)
9. Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfharinger, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream classification. Machine Learning **106**(9-10), 1469–1495 (2017)
10. Gomes, H.M., Read, J., Bifet, A., Barddal, J.P., Gama, J.: Machine learning for streaming data: state of the art, challenges, and opportunities. ACM SIGKDD Explorations Newsletter **21**(2), 6–22 (2019)
11. Gözüaçık, Ö., Büyükçakır, A., Bonab, H., Can, F.: Unsupervised concept drift detection with a discriminative classifier. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2365–2368 (2019)
12. Harries, M., Wales, N.S.: Splice-2 comparative evaluation: Electricity pricing (1999)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Kim, Y., Park, C.H.: An efficient concept drift detection method for streaming data under limited labeling. IEICE Transactions on Information and systems **100**(10), 2537–2546 (2017)
15. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information theory **37**(1), 145–151 (1991)
16. Montiel, J., Read, J., Bifet, A., Abdessalem, T.: Scikit-multiflow: A multi-output streaming framework. The Journal of Machine Learning Research **19**(1), 2915–2914 (2018)
17. Page, E.S.: Continuous inspection schemes. Biometrika **41**(1/2), 100–115 (1954)
18. Pinto, F., Sampaio, M.O., Bizarro, P.: Automatic model monitoring for data streams. arXiv preprint arXiv:1908.04240 (2019)
19. dos Reis, D.M., Flach, P., Matwin, S., Batista, G.: Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1545–1554 (2016)
20. Yu, S., Wang, X., Principe, J.C.: Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels. arXiv preprint arXiv:1806.10131 (2018)
21. Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (2018)
22. Žliobaite, I.: Change with delayed labeling: When is it detectable? In: 2010 IEEE International Conference on Data Mining Workshops. pp. 843–850. IEEE (2010)