Quantifying Concept Drift using Hellinger Distance Based Drift Detection

Thomas Boot
Eindhoven University of Technology
t.boot@student.tue.nl

Abstract—Concept drift is a challenging problem in the context of online learning. How one adapts to concept drift depends on the nature of the detected drift, such as the drift magnitude or duration. While quantifying the characteristics of this drift is thus crucial, there currently exist no drift detectors that manage to do so. This research proposes an approach to quantify the drift magnitude during drift detection. Our approach makes use of the Hellinger Distance Drift Detection Method (HDDDM), which computes the Hellinger distance between points in time in order to measure the drift magnitude between the underlying data distributions. Two adaptations that condition the distance computation on the class labels are suggested as improvements on the baseline HDDDM approach. The first adaptation conditions the distance on the class that experiences the largest degree of change. The second adaptation makes use of a weighted sum of conditional distances. All drift detectors are evaluated on different datasets with varying degrees of abrupt and gradual concept drift. Results especially show the promise of the first adaptation when quantifying concept drift, providing clear visual indication on the type of drift occurring.

Index Terms—Concept Drift, Drift Magnitude, Hellinger Distance, HDDDM.

1 Introduction

Real world data tends to change over time [9]. As a result, detecting change in the underlying distribution of the data has increasingly received attention in data mining and machine learning applications. Such a change in a data distribution - or *concept* - is generally referred to as *concept drift*. For an algorithm to effectively generalize on future data, it is highly relevant that it is able to adapt to concept drift [4]. For example, consider the case of anomaly detection for the purpose of monitoring credit card fraud. Credit card transactions are being monitored in order to identify potentially fraudulent behavior. A consumer's purchasing habits may however change (e.g. after winning the lottery). Therefore, some transactions that were previously not considered as anomalies might now be unusual, and vice versa. It is thus important for an algorithm that monitors potential fraud to be adaptive to such a change in transaction behavior. Similar needs for adaptive algorithms arise in multiple other applications such as recommender systems, forecasting models or crime prediction [9]. That is why nowadays, there is an increasing amount of research spent towards developing drift detection algorithms [3, 5, 15].

An important consideration when dealing with concept drift is how one adapts to the detected drift. Different adaptation strategies exists, such as retraining or updating the model upon drift detection. However, the preferred adaptation strategy depends on the nature of the drift itself [3]. It is therefore not sufficient to simply detect drift. It is also necessary to identify relevant characteristics of the concept drift. Even though there exist many drift detection algorithms [15], there exist none that we know of that provide detailed information on such characteristics. There has been some research on quantifying concept drift through the means of *Drift Maps* [14]. However, this kind of research is reactive and assumes all data is know beforehand. This is an assumption that often does not hold as concept drift is usually associated with streaming data. In such an online setting, new data keeps being added to the stream (either in batches or one at a time).

This paper proposes a new approach towards quantifying concept drift. Our approach makes use of a state of the art drift detector - called the *Hellinger Distance Drift Detection Method* or *HDDDM* - that computes the Hellinger Distance between timeframes in order to determine whether data distributions at different timestamps are significantly different [4]. Extracting these distances from the drift detector can be used to quantify the drift magnitude, which is relevant when deciding on an appropriate adaptation strategy. This approach is implemented and tested on various artificial datasets with different types of drift.

The remainder of this paper is structured as follows. First, we formalize the notion of concept drift and its characteristics in Sect. 2. Related work on quantifying concept drift is discussed in Sect. 3. Sect. 4 elaborates on the setup of our drift quantification algorithm. This setup is tested on various artificial datasets in Sect. 5, after which conclusions are drawn in Sect. 6.

• *Thomas A. Boot is with Eindhoven University of Technology. E-mail: t.boot@student.tue.nl. Student number: 0988095.*

2 Problem Description

Online learning is concerned with the training of models without having all the data available beforehand. New data is typically added to the data stream in batches or one at a time. The underlying distribution of this data may change over time, also known as *concept drift*. We start by formally defining the notion of concept drift.

Consider a stream generated from a joint distribution over random variables Y and $X = X_1, ..., X_n$, where $y \in Y$ are the class labels and $\mathbf{x} \in X$ the attribute values. The joint probability distribution at a specific time t is denoted as $P_t(X,Y)$. Following Gama et al's [6] definition, such joint probability distribution is also referred to as a concept.

$$Concept = P_t(X,Y) \tag{1}$$

Concept drift is known to occur when distributions change between timestamps t and u,

$$P_t(X,Y) \neq P_u(X,Y) \tag{2}$$

or similarly between timeframes $[t,u]$ and $[v,w]$ (when dealing with batches):

$$P_{[t,u]}(X,Y) \neq P_{[v,w]}(X,Y). \tag{3}$$

Concept drift can be further characterized according to the work of Webb et al [13]. Two particularly interesting characteristics concern the *Drift Magnitude* and *Drift Duration* as these generally have the most impact on learner selection and adaptation [3].

Drift Magnitude is the distance between the initial and resulting concept over the period of drift. The magnitude of drift between times t and u is

$$Magnitude_{[t,u]} = D(t,u) \tag{4}$$

where D is a distance function that quantifies the difference between both concepts.

Drift Duration refers to the elapsed time over which a period of drift occurs. The duration of a drift between timestamps t and u is

$$Duration_{[t,u]} = u - t. \tag{5}$$

Both the magnitude and duration enable the distinction of concept drift into *abrupt (sudden) drift* and *gradual drift* [13].

Abrupt (sudden) drift is a concept drift occurring over a small time period γ (possibly instantaneous):

$$Duration_{[t,u]} < \gamma. \tag{6}$$

Gradual drift occurs when the magnitude of the drift remains smaller than a maximum difference μ between concepts over a longer period of time:

$$\forall_{v \in [t,u]} D(t,u) < \mu. \tag{7}$$

In order to determine a suitable adaptation strategy when confronted with concept drift, it is crucial to understand the characteristics of this drift. Even though there exist multiple drift detection algorithms (e.g. EDDM [2]) which alert the model when drift is present, there are no algorithms that include the magnitude or duration of the drift in order to provide more accurate information on the type of drift that occurred. This information is relevant as abrupt drift might require retraining the entire model, while it might suffice to refine the model in case of gradual drift [3]. The scope of our research is primarily concerned with quantifying the drift magnitude. Quantifying the drift duration would be valuable future research.

3 Related Work

Even though there exists no algorithm that effectively quantifies drift magnitude that we know of, there has been some interesting research that is useful for this purpose.

3.1 Hellinger Distance Drift Detection Method

Ditzler and Polikar [4] have proposed a feature based drift detection algorithm that uses the Hellinger distance to detect gradual or abrupt drift in a given data distribution. This Hellinger Distance Drift Detection Method (HDDDM) computes the Hellinger Distance between batches of data and maintains an adaptive threshold to indicate whether the magnitude of distance between a new distribution and baseline distribution exceeds acceptable levels.

Even though this approach is suitable for other distance metrics, the authors specifically argue the advantage of the Hellinger distance as this is a symmetric distance measure and does not make any assumptions on the distribution of the data. Also, the Hellinger distance is a measure of distributional divergence, which makes it suitable to measure a change between distributions of consecutive time stamps.

The pseudocode for the HDDDM algorithm as proposed by Ditzler and Polikar [4] is presented in Algorithm 1. Data is separated into batches, where D_t indicates the batch at timestamp t. The algorithm initializes by setting $D_\lambda = D_1$, where λ indicates the last batch in which drift was detected. For each batch, a histogram is created. The Hellinger distance $\delta_H(t)$ between histograms is computed using Equation 8 below. In this equation, d refers to the dimensionality of the data and $P_{i,k}(Q_{i,k})$ is the frequency count in bin i of the histogram corresponding to the histogram $P(Q)$ of feature k. In other words, the joint probability distribution at D_t is approximated through the proportions of the different bins per feature.

$$\delta_H(t) = \frac{1}{d} \sum_{k=1}^{d} \sqrt{\sum_{i=1}^{b} \left(\sqrt{\frac{P_{i,k}}{\sum_{j=1}^{b} P_{j,k}}} - \sqrt{\frac{Q_{i,k}}{\sum_{j=1}^{b} Q_{j,k}}} \right)^2} \tag{8}$$

Algorithm 1 HDDDM

Input: Training Data $D_t = \{\mathbf{x}^{(t)} \in X; y^{(t)} \in Y\}$, presented in batches corresponding to a joint probability distribution $P_t(X,Y)$ where $t = 1, 2, ...$
Initialize: $\lambda = 1$ and $D_\lambda = D_1$

for $t = 2, 3, ...$ **do**

1. Generate two histograms P and Q from D_t and D_λ respectively. Each histogram has $b = \lfloor \sqrt{N} \rfloor$ bins, where N is the cardinality of D_t.
2. Calculate the Hellinger distance between P and Q as given by Equation 8. Call this $\delta_H(t)$.
3. Compute the difference in Hellinger distance:
$$\varepsilon(t) = \delta_H(t) - \delta_H(t-1)$$
4. Update the adaptive threshold:
$$\hat{\varepsilon} = \frac{1}{t - \lambda - 1} \sum_{i=\lambda}^{t-1} |\varepsilon(i)|$$
$$\hat{\sigma} = \sqrt{\frac{\sum_{i=\lambda}^{t-1} (|\varepsilon(i)| - \hat{\varepsilon})^2}{t - \lambda - 1}}$$
5. Compute $\beta(t)$ using the computed measures:
$$\beta(t) = \hat{\varepsilon} + \gamma \hat{\sigma}$$
6. Determine if drift is present.

if $|\varepsilon(t)| > \beta(t)$ **then**
  $\lambda = t$
  Reset D_λ by setting $D_\lambda = D_t$
  Indicate change is detected.
else
  Update D_λ with $D_t \rightarrow D_\lambda = \{D_\lambda, D_t\}$
end if

end for

The algorithm continues by computing the difference in Hellinger distances between batches and using this difference to maintain an adaptive threshold. This threshold, $\beta(t)$, is based on a standard deviation around the average difference in Hellinger distance. The amount of deviation from the mean allowed can be maintained through γ, which is some positive constant. If drift is detected, the algorithm resets. If not, the reference window is updated by accumulating the current window and the previous reference window.

The authors proceed by testing the algorithm in various experiments, both using real world and artificial data. Preliminary results have shown that HDDDM can improve the performance of an incremental learning algorithm for both gradual and abrupt drift. The use of this algorithm proves interesting for our purpose as it provides a quantitative measure to maintain how "different" the distributions between batches are. When drift is detected, $\delta_H(t)$ can be extracted and used as an quantitative measure for the magnitude of the drift occurred.

3.2 Quantifying Gradual Drift

Quantifying gradual drift is a notoriously challenging problem. In their work, Sarnelle et al. [10] propose a classifier, COMPOSE, that performs well under gradual drift. In order to validate this assumption, they attempt to define a measure that formally quantifies gradual drift. They propose two metrics, one that represents the normalized class separation and one that represents the drift classification risk.

Their work assumes a binary problem, where X_t is the set of obser-

vations belonging to class 1 at time t and Y_t the set of observations belonging to class 2 at that time. They define a distance metric $D(X,Y)$ (e.g. Hellinger distance) that computes the distance between the distributions of both classes. Using this notation, the define the *Normalized Class Separation (NCS)* based drift as:

$$\Delta(t) = \frac{\max(D(X_t, X_{t-1}), D(Y_t, Y_{t-1}))}{D(X_{t-1}, Y_{t-1})} \tag{9}$$

In other words, they argue drift can be quantified by evaluating which class has drifted more significantly over time, normalized by the prior distance that existed between both classes. One drawback of this metric is that it merely measures the amount of drift occurred, disregarding the direction of the drift. This might be relevant for classification purposes as it is significantly more difficult to distinguish classes that drift towards each other than classifying classes that drift away from each other. The authors therefore propose a second metric, the *Drift Classification Risk*, that takes the direction of the drift into account. However, as we are more interested in quantifying drift rather than improving classifier performance, this second metric falls beyond the relevant scope of this research.

It is important to notice that the work of Sarnelle et al. aims to solve a different challenge than we do. Their scope was primarily concerned with assessing whether drift is limited enough for their classifier to still perform well. Their experimental setup primarily assesses how large $\Delta(t)$ can become before their classifier deteriorates. We are, however, concerned with a more general setting in which we aim to quantify the drift magnitude in order to provide insight in the characteristics of the drift in question. Even so, the notion of measuring drift by evaluating the drift between classes might provide interesting insights for our purpose.

3.3 Drift Maps

Webb et al. [14] propose the use of *Drift Maps* for the description and analysis of concept drift. Similar to Ditzler and Polikar, they argue the use of the Hellinger distance as a measure of drift magnitude. They also argue the relevance of measuring marginal drift magnitudes, which allows for the analysis of concept drift in smaller attribute subspaces of the data. They then proceed by mapping the conditional marginal covariate drift and conditional class drift in order to provide detailed insights on the concept drift on a fine level of granularity.

Even though the work of Webb et al. shows great results and relevance in quantifying concept drift, a large drawback remains in the fact that drift mapping differs greatly from drift detection. The former aims to describe in detail the nature of the drift that has occurred, while the latter aims to identify when drift is occurring. While drift detection is often used in online learning, drift maps are mainly used for data analysis purposes and assume all the data, including when drift has occurred, is know beforehand. To that extent, drift maps are not directly applicable in the context of online learning.

4 Approach

This research aims to develop a drift detector that not only alerts the user when drift occurs, but also provides information on the drift magnitude. We propose an implementation of the HDDDM algorithm proposed by Ditzler and Polikar [4] and investigate whether this algorithm can be modified to provide information on the drift magnitude. Furthermore, two alternative adaptations of the HDDDM algorithm are proposed based on the work of Sarnelle et al. [10] and Webb et al. [14] respectively.

This section starts by discussing the HDDDM implementation. Next, the motivation for developing two adaptations of this algorithm is introduced. This section concludes by explaining the experimental setup that is used for the evaluation of the three algorithms' performances.

4.1 Hellinger Distance Drift Detection Method

The primary approach for quantifying drift magnitude is conducted by implementing Algorithm 1 by Ditzler and Polikar [4]. This algorithm is not yet implemented in established libraries (e.g. scikit multiflow), so it has been manually implemented [1].

The data is preprocessed by discretizing all numerical features into bins. Subsequently, the binned data is divided into batches. For every new batch D_t, the HDDDM algorithm computes the Hellinger distance between that batch and the reference batch D_λ as defined in Equation 8. This is done by computing the intermediate Hellinger distance per feature and averaging this distance over the different features. The computation of the intermediate Hellinger distances is conducted by maintaining the proportions of the different bins per feature. After computing the Hellinger distance, the drift detector proceeds as in Algorithm 1 in order to determine if drift has occurred in that batch. When drift is detected, the drift detector resets. Adjusting the γ parameter allows the user to define what magnitude of change is considered severe enough to trigger the drift detector. The implemented algorithm also allows for warnings whenever the magnitude of change gets reasonably close to the threshold $\beta(t)$ based on a user-defined measure what is considered "reasonably close".

The Hellinger distances $\delta_H(t)$ computed by the HDDDM algorithm are stores separately as a measure of the magnitude of change in that batch. This allows the drift detector to not only warn the user when drift is present, but also output information on the magnitude of that drift.

4.2 Alternative Algorithms

One possible drawback from the HDDDM algorithm is that there is no distinction made between the features $\mathbf{x} \in X$ and the class labels $y \in Y$. The class variable is essentially regarded as one of the features during the calculation of the Hellinger distance. This could lead to misleading results, as a significant drift in class distributions $P(Y)$ could be overlooked by a fairly consistent covariate distribution $P(X)$ within the features. Even though this would significantly decrease a classifier's performance, drift would not be detected.

In order to prevent such a drawback, inspiration is drawn from previous studies. Both Sarnelle et al. [10] and Webb et al. [14] have proposed a measure that conditions the distance computation based on the class values. Given their different approaches on the matter, we suggest two different adaptations on the HDDDM algorithm.

4.2.1 Adaptation 1

The first adaptation is inspired by the NCS metric developed by Sarnelle et al. [10] in Equation 9. They compute some distance measure within each class separately and subsequently identify the class in which the most drift has occurred. A similar approach is adopted in our work. First of all, the class variable is separated from the other features. Within each batch D_t, we condition the data based on their class label, leading to separate datasets $D_{t \mid y}$ for each class $y \in Y$. For each class y, the Hellinger distance is computed similar to Equation 8. The final Hellinger distance is then computed as follows:

$$\delta_H(t) = \max(\delta_H(t \mid y)), \tag{10}$$

or in a binary situation:

$$\delta_H(t) = \max(\delta_H(t \mid y_1), \delta_H(t \mid y_2)). \tag{11}$$

The remainder of this adaptation functions in a similar fashion as the HDDDM algorithm. Note that, in contrast to the NCS measure, we do not normalize the Hellinger distance. That is because normalizing occurs in consecutive steps in the HDDDM algorithm.

4.2.2 Adaptation 2

The second adaptation is based on the notion of marginal drift magnitude as described by Webb et al. [14], which argues the need for measuring drift in different subspaces of the data. They introduce the conditional marginal covariate drift $\sigma^{X \mid Y}_{t,u}$:

[1] Some experimental code [8] has been found online and is used as baseline for the implementation of HDDDM.

$$\sigma_{t,u}^{X \mid Y} = \sum_{y \in Y} \left[\frac{P_t(y) + P_u(y)}{2} D(t, u \mid y) \right], \tag{12}$$

where $P_t(y)$ is the probability of class y occurring at time t and $P_u(y)$ the probability of that class occurring at time u. $D(t, u \mid y)$ is some distance metric between t and u, conditioned on the class value. In other words, they argue the use of conditional marginal drift computation by taking the weighted sum of the distances between the conditional probability distributions. A similar idea is adopted in our work. Each batch D_t is conditioned on the occurring class labels, leading to separate datasets $D_{t \mid y}$. The Hellinger distance is computed using the following equation:

$$\delta_H(t) = \sum_{y \in Y} \left[\frac{P_t(y) + P_\lambda(y)}{2} \delta_H(t \mid y) \right], \tag{13}$$

where $P_\lambda(y)$ refers to the probability of y occurring in the reference batch D_λ. The remainder of this adaptation follows the same approach as HDDDM.

4.3 Experimental Setup

This section describes how the different drift detectors have been evaluated and compared. The experimental setup is conducted by creating an online learning environment and evaluating the ability of the different detectors to help a classifier handle drifting data streams. The Gradient Boosting (GBM) classifier [12] is used from the scikit-learn library, as it is known to react effectively to drifting environments [3]. The hyperparameters are set to default. Different datasets with varying types of drift are used. The data is separated into batches of $n = 1000$ instances. The initial batch D_1 is used for model training, subsequent batches are used for model inference. Whenever the drift detector detects drift in a batch D_t, the classifier retrains on that batch. The γ parameter is initialized at 1.5. The warning level is set to 0.8, meaning that $|\varepsilon(t)| > 0.8\ \beta(t)$ would already alert the user with a drift warning.

The aim of the experimental setup is to evaluate whether the different drift detectors can correctly identify the drift points in the dataset and to investigate whether the detectors can correctly distinguish between different types and magnitudes of drift.

4.3.1 Datasets Used

The drift detectors have been evaluated on seven artificial datasets with varying degrees of concept drift, including gradual, abrupt and mixed (gradual and abrupt) drift. Within each dataset, concept drift is inserted by altering the underlying distributions of the data. All datasets are synthetically generated using the River library and assume a binary classification problem. The stream generators used are described below. An overview of the used datasets is given in Table 1.

- *Streaming Ensemble Algorithm (SEA)* generates data streams based on four underlying classification functions [11]. Data streams of 500 000 instances are created with 3 numerical features. Abrupt drift is created at instance 250 000 by changing between classification functions. The drift window *width*, referring to the timeframe in which drift occurs, is set to $width = 1$. Three varieties on this dataset are generated with different drift magnitudes (high, middle and low abrupt drift) by altering which of the four classification functions are used to insert concept drift.

- *Hyperplane* generates a rotating hyperplane used for binary classification in a d-dimensional space [7]. Gradual drift can be inserted by changing the rotation and position of the hyperplane over time. Three datasets of 500 000 instances with 10 features are created. Different magnitudes of gradual drift are used (10%, 1% and 0.1% for the three datasets respectively) on the dataset, indicating the magnitude of change between each instance. Drift is inserted throughout the entire dataset: *width* = 500 000. A σ of 0.1 is added, indicating the probability that the direction of change is reversed.

- *Agrawal* is a stream generator that produces a stream containing nine features, six numeric and three categorical [1]. There are ten classification functions that generate binary class labels from these features. Changing the classification function allows for concept drift in the stream. A dataset of 1 000 000 instances is created with mixed drift. Abrupt drift is inserted at instance 500 000, with $width = 1$. Two gradual drifts are also created at windows [200 000, 300 000] and [700 000, 800 000], both with *width* = 100 000 (i.e. a drift window of 100 000 instances).

Table 1: Synthetic Datasets used for Experimental Setup.

Generated Datasets		
Dataset	**Instances**	**Drift Type**
SEA - High	500 000	High Abrupt Drift
SEA - Middle	500 000	Middle Abrupt Drift
SEA - Low	500 000	Low Abrupt Drift
Hyperplane - 10%	500 000	High Gradual Drift
Hyperplane - 1%	500 000	Middle Gradual Drift
Hyperplane - 0.1%	500 000	Low Gradual Drift
Agrawal	1 000 000	Mixed Drift

5 Results

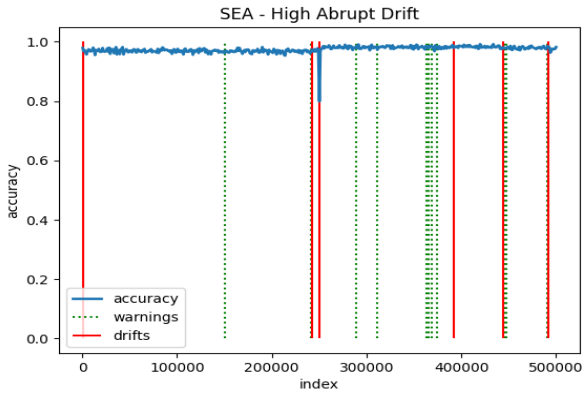
We evaluate the HDDDM algorithm, as well as both its adaptations, on the different datasets. For each dataset, we investigate whether the drift detector correctly identifies the drift and whether it is able to quantify the drift magnitude. The most interesting results appear when dealing with abrupt drift. This sections will primarily focus on these results. Afterwards, results for gradual and mixed drift are briefly discussed. We conclude the results by discussing limitations of our approach.

5.1 Abrupt Drift

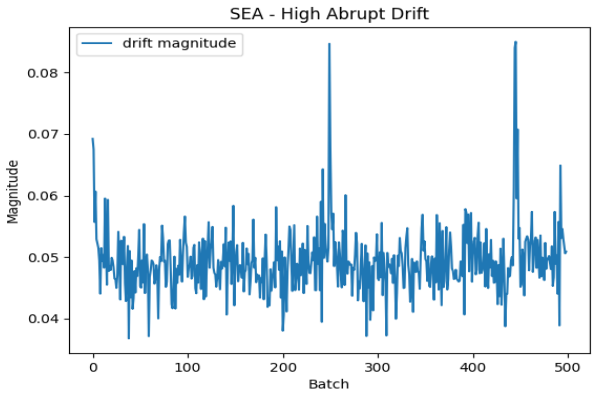
Fig. 1 shows the results of the different drift detectors on high abrupt drift. The plots on the left show the classifier accuracy. Red lines indicate detected drift, green dotted lines indicate drift warnings. The plots on the right visualize the corresponding Hellinger distance magnitudes. Recall that drift has been inserted in the middle of the dataset. HDDDM correctly identifies this drift (visible in figure Fig. 1a), which is also apparent through a significant rise in drift magnitude at that point in Fig. 1b. The Hellinger distance $\delta_H(t)$ remains fairly constant for all t at around 0.05, except for the batch in which drift occurs ($\delta_H(t) \approx 0.85$). This sudden rise in drift magnitude also confirms the suspicion on the drift being abrupt. Interestingly, there is one more significant rise in Hellinger distance around batch 450. While there is no drift at this point, this might be due to random noise in the underlying covariate distribution $P_t(X)$.

Both adaptations provide even more satisfying results. The first adaptation not only correctly identifies abrupt drift, but also visually shows a shift in the underlying distributions in Fig. 1d. While the average Hellinger distance before batch 250 lies around 0.10, it rises to an average of 0.12 after abrupt drift. By Equation 11, this means that the Hellinger distance within at least one of the classes has become consistently higher and the underlying distribution thus likely has changed. This provides additional reassurance that the detected drift at $t = 250$ indeed was an actual drift instead of noise. The second adaptation provides additional benefits by being very robust against noise. Fig. 1e and Fig. 1f show that drift is essentially only detected at the moment in time where drift actually occurred. Any rise in drift magnitude thus indicates abrupt drift with a higher degree of certainty. This is likely due to Equation 13 taking a weighted sum over conditional distances, making it more robust against e.g. class imbalance.

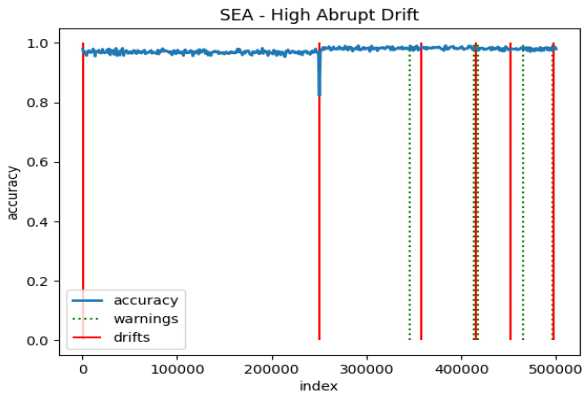
In short, all three drift detectors correctly identify high abrupt drift and allow the classifier to recover quickly. For all three drift detectors, there is a clear rise in Hellinger distance magnitude when drift occurs, confirming that this provides information on the drift magnitude. Since all three detectors use different measures, and are thus quantified on a different scale, the drift magnitudes are not directly comparable.



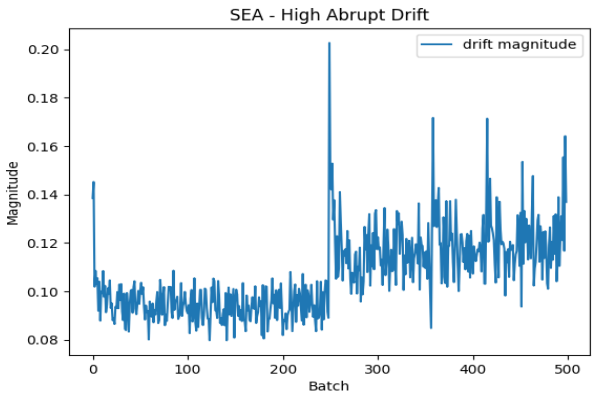
(a) HDDDM - Drift Detection



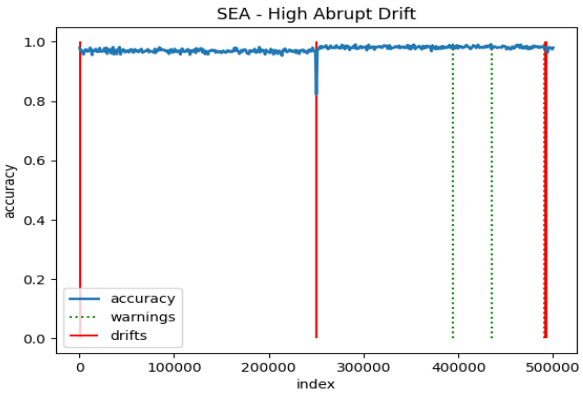
(b) HDDDM - Drift Magnitude



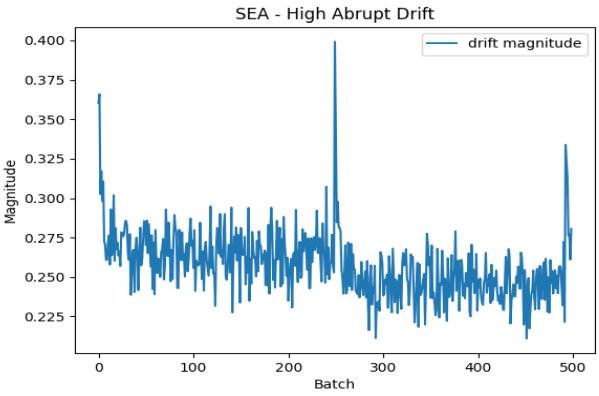
(c) Adaptation 1 - Drift Detection



(d) Adaptation 1 - Drift Magnitude



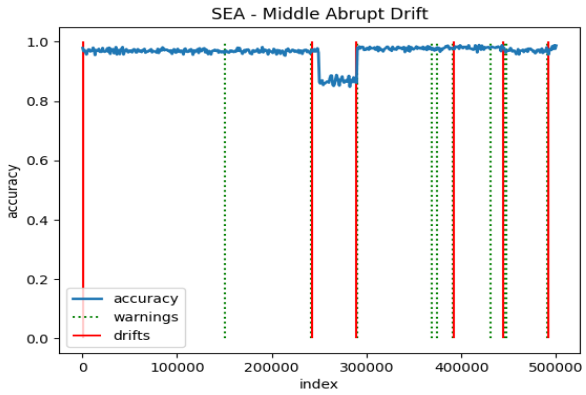
(e) Adaptation 2 - Drift Detection



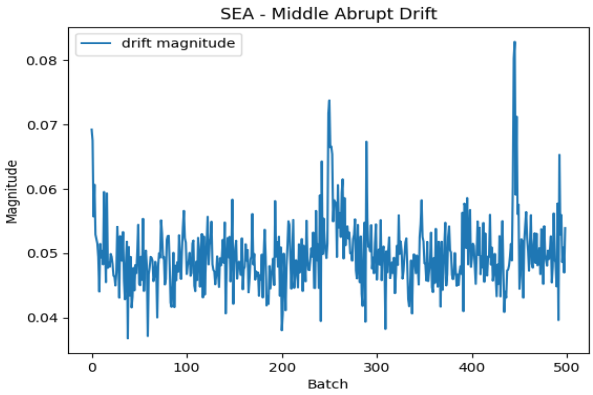
(f) Adaptation 2 - Drift Magnitude

Fig. 1: Drift Detector Performance on High Abrupt Drift

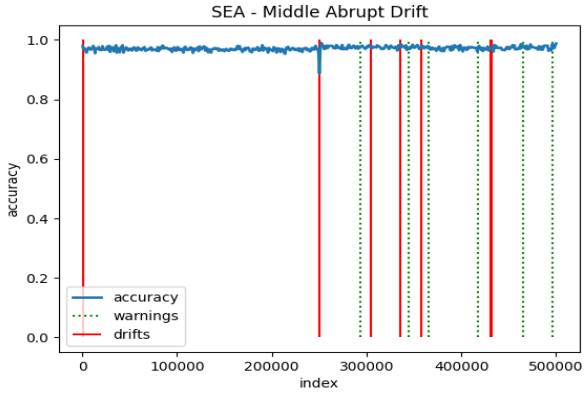
Additional experiments have been conducted on data with middle abrupt drift in order to compare the results of the drift detectors for different drift magnitudes. Fig. 2 shows the results from the drift detectors on middle abrupt drift. Similar to high abrupt drift, drift is inserted at batch $t = 250$. Fig. 2b shows that HDDDM still identifies a rise in $\delta_H(t)$ at $t = 250$. However, this increase is not high enough. It takes the algorithm longer to surpass the threshold and trigger the drift detector, leading to the classifier needing longer to recover. Adaptation 1 is able to identify the abrupt drift. Here too, a slight shift in probability distributions is visible in Fig. 2d: $\delta_H(t) \approx 0.10 \; \forall \; t \in [0, 249]$ compared to $\delta_H(t) \approx 0.11 \; \forall \; t \in [251, 500]$. The detectors also identifies a reasonable amount of noise. The second adaptation is not able to identify the drift, but does provide a warning. There is a clear increase in Hellinger distance at the drift point in Fig. 2f. However, this is not large enough to trigger the detector. Consequently, batches keep being added to the reference batch D_λ, which means that an increasing proportion of the new distribution keeps being added to the reference batch. Since this approach takes into account a weighted sum during distance computation, the Hellinger distance keeps decreasing and drift is never triggered. The classifier does not recover. A possible manner to overcome this issue is by adjusting the settings for γ. Doing so enables the drift detector to correctly identify the drift. However, it is important to notice this also makes the detector more sensitive to noise as it triggers for every small change in the Hellinger distance.



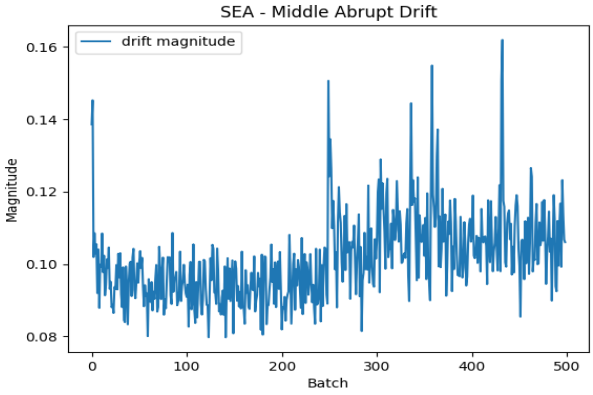
(a) HDDDM - Drift Detection



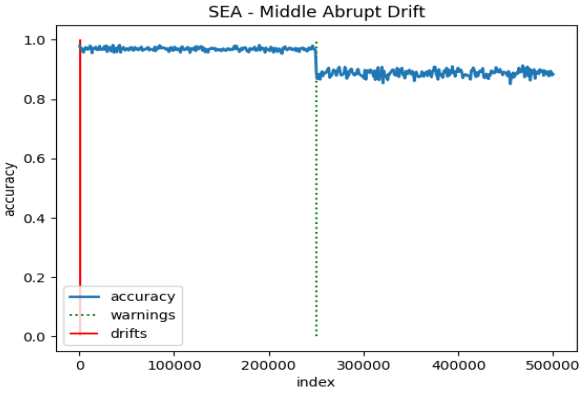
(b) HDDDM - Drift Magnitude



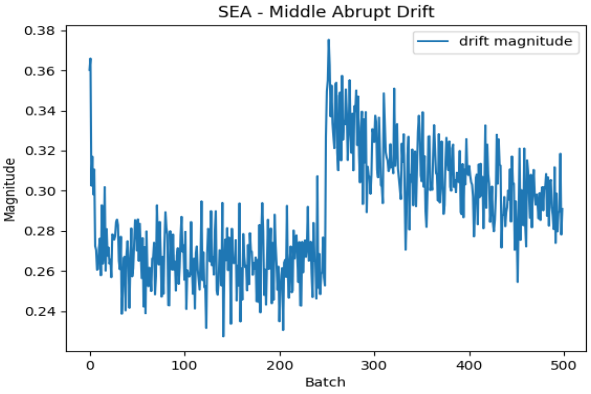
(c) Adaptation 1 - Drift Detection



(d) Adaptation 1 - Drift Magnitude



(e) Adaptation 2 - Drift Detection



(f) Adaptation 2 - Drift Magnitude

Fig. 2: Drift Detector Performance on Middle Abrupt Drift

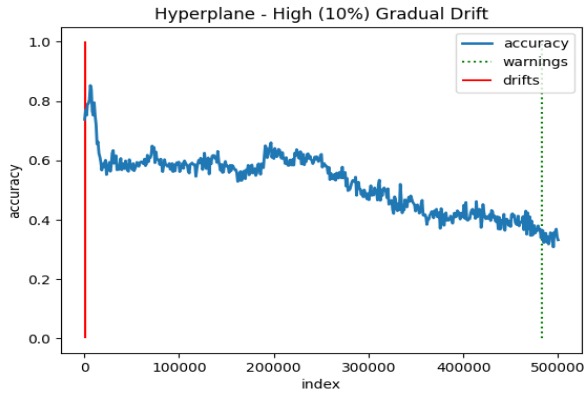
Results for low abrupt drift are omitted as none of the drift detectors are able to identify the drift. However, results are highly similar as for middle abrupt drift: HDDDM needs time before drift is triggered, adaptation 1 identifies a small shift in distributions and adaptation 2 is not able to recover at all. However, as the drift in low abrupt drift is so small, the performance of the classifier hardly deteriorates.

Three conclusions can be drawn from the results on datasets with abrupt drift:

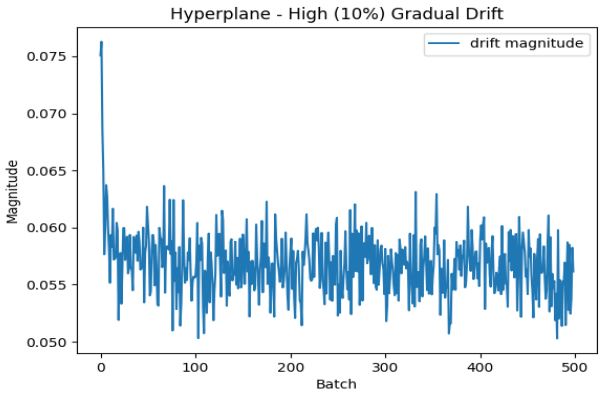
1. Adaptation 1 works best on abrupt drift, being able to detect drift that is reasonably large and showing a shift in data distributions before and after the drift. HDDDM works reasonably well but needs time to recover as drift gets smaller. Adaptation 2 is robust against noise when dealing with high drift. However, this detector has difficulty recovering when drift gets smaller.

2. Regardless of drift actually being detected, all drift detectors are able to visually identify abrupt drift. This is visible through a clear rise in the drift magnitude at the moment where drift occurs and sometimes a shift in the drift magnitude before and after drift.

3. All drift detectors provide some quantitative ordering of the drift magnitudes. For example, HDDDM clearly identifies high abrupt drift (Fig. 1b shows $\delta_H(t) \approx 0.85$) as being larger than middle abrupt drift (Fig. 2b shows $\delta_H(t) \approx 0.75$).

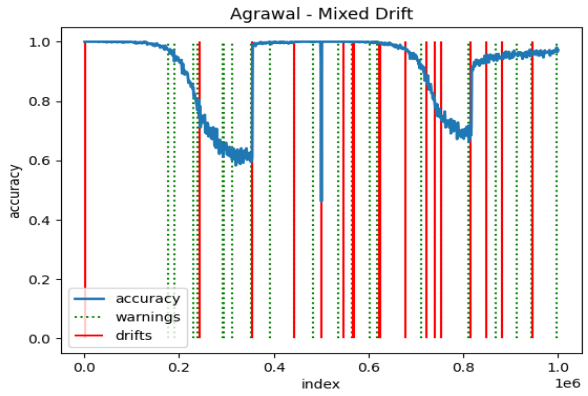


(a) HDDDM - Drift Detection

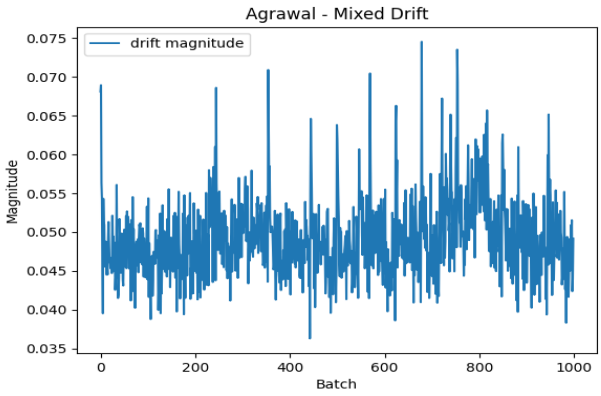


(b) HDDDM - Drift Magnitude

Fig. 3: HDDDM Performance on High Gradual Drift



(a) HDDDM - Drift Detection



(b) HDDDM - Drift Magnitude

Fig. 4: HDDDM Performance on Mixed Drift

5.2 Gradual Drift

The results from applying HDDDM on high gradual drift are shown in Fig. 3. It can be seen that the drift detector is not able to identify gradual drift and the classifier strongly drops in performance over time (Fig. 3a). This is due to the fact that gradual drift is present throughout the entire dataset in a consistent manner. As can be seen in Fig. 3b, the drift magnitude remains consistent over time. Since HDDDM relies on the detection of distinct changes in the Hellinger distance, the detector is never triggered. Valuable future research would include experimenting with varying drift windows in order to assess whether this alters the behavior of HDDDM.

Results from the two adaptations show no improvement, as they equally rely on significant changes within this distance (i.e. the classifier's performance remains identical to Fig. 3a). Adaptation 1 does show some interesting visual properties, which are discussed later. Since none of the drift detectors are able to handle high gradual drift, results for middle and low gradual drift are omitted.

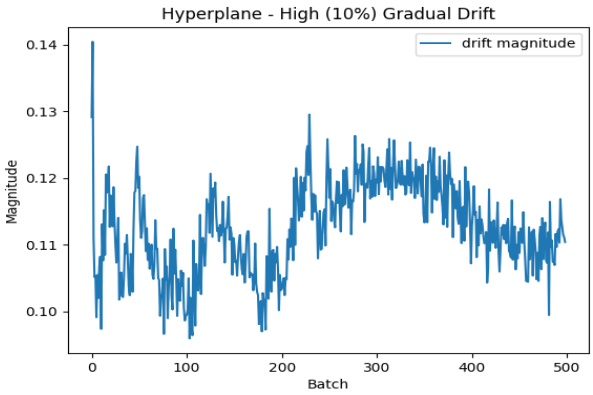
5.3 Mixed Drift

The performance of HDDDM on mixed drift is presented in Fig. 4. Recall that gradual drift occurs between batches $t \in [200, 300]$ and $t \in [700, 800]$ and abrupt drift occurs at batch $t = 500$. It can be seen that the drift detector is able to eventually detect the gradual drift and immediately identifies the abrupt drift. Even so, the algorithm is not able to provide valuable insights in the type of drift that occurred. Fig. 5b shows a lot of noise, with large increases in the drift magnitude occurring at arbitrary points in time. The drift magnitude when abrupt drift occurs is smaller than most of the other peaks, making it difficult for the algorithm to identify that point in time as the moment where abrupt drift occurs. A slight increase in drift magnitude is visible during the periods of gradual drift. However, due to the large amount of noise, is not able to definitively identify this as the period in which gradual drift occurred. If noise can be reduced in future research, insights in the drift magnitude would be a lot more valuable.

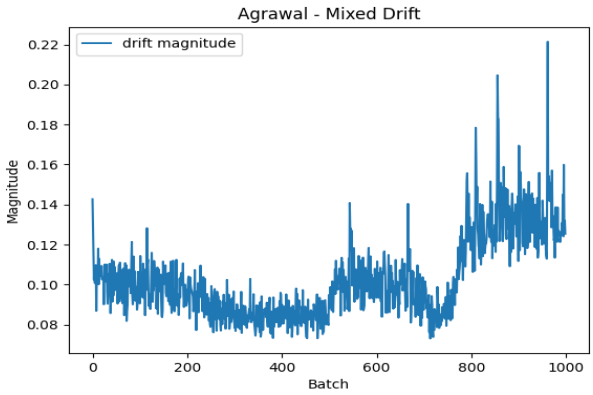
The results for the two adaptations are omitted as they fail to identify any of the relevant drift points. However, similar as for gradual drift, adaptation 1 shows some visual properties. We therefore briefly reflect on the performance of this adaptation during gradual and mixed drift.

5.3.1 Insights from Adaptation 1

Fig. 5 shows the magnitude plots of adaptation 1 during gradual and mixed drift. It can be seen that for gradual drift, the drift magnitude fluctuates. Even though these fluctuation currently do not provide any valuable information, they provide some indication that the underlying distribution of the data is changing. Since this adaptation conditions the Hellinger distance on the class with the highest degree of change, these fluctuations mean that the underlying distribution within at least one class continuously changes. However, these fluctuations are too small to actually trigger the drift detector. For mixed drift, Fig. 5b shows how this approach is less prone to noise than the baseline HDDDM approach. The figure shows a gradual change in the drift magnitude during the time periods in which gradual drift occurs. Again, this gradual change is too small for the drift detector to trigger. This also



(a) Adaptation 1 - Drift Magnitude on Gradual Drift



(b) Adaptation 1 - Drift Magnitude on Mixed Drift

Fig. 5: Drift Magnitude of Adaptation 1 on Gradual and Mixed Drift

explains why abrupt drift is not detected: the dataset induces abrupt drift by resetting its distribution back to its initial distribution. Since the gradual drift was too small to notice, resetting this drift would not be noticed either. Even though the drift detector fails to detect both the gradual and mixed drift, we deliberately decided to include these plots as they show some visual promise to identify gradual drift in a data stream. However, improving this approach in order to provide any quantitative, significant information on gradual drift would require considerable future research.

5.4 Limitations

The results show significant promise, especially dealing with abrupt drift. Overall, the first adaptation on the baseline HDDDM algorithm provides the most promising results, given its visual interpretation of concept drift. Even so, a number of limitations occurred during the experiments.

First of all, regardless of the drift detector chosen, the results remains intuitive and visual. For example, high abrupt drift is identified by a significant increase in $\delta_H(t)$, but does not provide any information on the drift magnitude in an absolute sense. Ideally, one would design some method to benchmark drift magnitudes in order to provide conclusive results without the need of visual aid: e.g. any drift magnitude above $\delta_H(t) = x$ is considered as severe drift.

Another important limitation is the applicability of our approach on gradual drift. HDDDM, as well as its adaptations, require a change in drift magnitude that is large enough to surpass the threshold $\beta(t)$. Hence, none of the drift detectors managed to identify drift that was gradual in nature. HDDDM did detect some drifts on the mixed drift dataset, but this might as well have been due to random noise. Adaptation 1 provides some visual insights in the drift magnitude during gradual drift, but does not actually detect any drift. Further research is needed to enable the quantification of gradual concept drift.

A third limitation is noise. Many of the results show detected drifts at points in time where no drift is present. While this appears less relevant, i.e. we rather detect too many drifts than not detecting drift when it is actually needed, this does interfere with the interpretation of the drift magnitude. For example, results on mixed drift show that HDDDM is not able to provide definite insights in the nature of the drift, even though the classifier does adapts to the drift. Adaptation 1 is not robust against noise either for the detector to only trigger when drift is actually present. Some noise might be resolved by careful parameter tuning (e.g. the γ parameter, batch size, ...). However, we hypothesize this is more likely a result of the current approach computing the distance between all covariate features, effectively including covariate drift in the drift detection. It would be insightful to investigate possibilities to more clearly separate the covariate drift from the class drift.

A final limitation appeared in the runtime of the drift detectors. Compared to established drift detectors (e.g. EDDM), our approach requires considerably more memory and time. This is due to the computational time of the intermediate Hellinger distances for every feature. Furthermore, the reference batch D_λ keeps increasing as long as drift is not detected. This is, however, a limitation that is inherent to using a distance measure for drift detection.

6 Conclusion

The main goal of this research has been to provide more insight in the characteristics of concept drift during drift detection. Providing this kind of insight is essential in an online learning setting, as the choice of adaptation strategy depends on the type of drift that occurred. Since characterizing drift is a broad field, our research restricted itself towards quantifying the drift magnitude during drift detection.

The proposed approach primarily draws inspiration from the Hellinger Distance Drift Detection Method (HDDDM) presented by Ditzler and Polikar [4]. This HDDDM algorithm computes the Hellinger distance between periods of time in order to quantify the degree of change between these time periods. The use of this Hellinger distance makes the drift detector suitable for quantifying drift magnitude, as it provides a measure for the difference in data distributions between time periods. One drawback of this approach is that it merely considers the class variable as one of the features during drift detection. To resolve this, two adaptations on the HDDDM algorithm are proposed that both condition the distance computation on the class labels. The first adaptation conditions the computation of the Hellinger distance on the class in which the degree of change is larger. The second adaptation computes the Hellinger distance based on a weighted sum between conditional distances.

Initial results have shown the promise of all three approaches in detecting and quantifying abrupt drift. Given the drift is severe enough, all approaches identify a distinct increase in Hellinger distance. The first adaptation shows the most promise as it visualizes a shift in probability distributions before and after abrupt drift. The drift detectors perform less promising on gradual and mixed drift, where each of the different approach either fails to detect drift or fails to show a significant difference in drift magnitude during the period of change. However, even though it fails to detect the correct drift points, the first adaptation provides some visual insights that could prove useful in identifying gradual drift. Given the overall results of this adaptation, we suggest to conduct further research into this approach.

To conclude, this research has provided a foundation towards quantifying concept drift. Suggestions for future research include reducing the dependence on visual intuition by benchmarking drift magnitudes (1), improving the applicability of this research on gradual drift (2) and investigating added benefits of separating covariate drift from class drift (3).

REFERENCES

[1] R. Agrawal, T. Imielinksi, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6), 1993.

[2] M. Baena-Garcia, J. del Campo-Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno. Early drift detection method.

[3] B. Celik and J. Vanschoren. Adaptation strategies for automated machine learning on evolving data.

[4] G. Ditzler and R. Polikar. Hellinger distance based drift detection for nonstationary environments. *IEEE*, 2011.

[5] S. Donoho. Early detection of insider trading in option markets. *Industry/Government Track Paper*, pages 420–429.

[6] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput Surv*, 46(4):1–37, 2014.

[7] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 97–106, 2001.

[8] Idriss. Github: drift_hdddm.

[9] N. Japkowicz and J. Stefanowski. *Big Data Analysis: new Algorithms for a new Society*, chapter 4. Springer, 2016.

[10] J. Samelle, A. Sanchez, R. Capo, J. Haas, and R. Polikar. Quantifying the limited and gradual concept drift assumption. *IEEE*, 2015.

[11] W. Street and Y. Kim. A streaming ensemble algorithm sea for large-scale classification. *7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 377 – 382, 2001.

[12] J. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren. Having a blast : meta-learning and heterogeneous ensembles for data streams. *15th IEEE International Conference on Data Mining*, pages 1003 – 1008, 2016.

[13] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964 – 994, 2016.

[14] G. I. Webb, L. K. Lee, B. Goethals, and F. Petitjean. *Analyzing concept drift and shift from sample data*, pages 1179 – 1199. Springer.

[15] M. M. W. Yan. Accurate detecting concept drift in evolving data streams. *ICT Express*, 6:332–338, 2020.