# Drift Detection Using Stream Volatility

David Tse Jung Huang[1](✉), Yun Sing Koh[1], Gillian Dobbie[1], and Albert Bifet[2]

[1] Department of Computer Science, University of Auckland, Auckland, New Zealand
{dtjh,ykoh,gill}@cs.auckland.ac.nz
[2] Huawei Noah's Ark Lab, Hong Kong, China
bifet.albert@huawei.com

**Abstract.** Current methods in data streams that detect concept drifts in the underlying distribution of data look at the distribution difference using statistical measures based on mean and variance. Existing methods are unable to proactively approximate the probability of a concept drift occurring and predict future drift points. We extend the current drift detection design by proposing the use of historical drift trends to estimate the probability of expecting a drift at different points across the stream, which we term the expected drift probability. We offer empirical evidence that applying our expected drift probability with the state-of-the-art drift detector, ADWIN, we can improve the detection performance of ADWIN by significantly reducing the false positive rate. To the best of our knowledge, this is the first work that investigates this idea. We also show that our overall concept can be easily incorporated back onto incremental classifiers such as VFDT and demonstrate that the performance of the classifier is further improved.

**Keywords:** Data stream · Drift detection · Stream volatility

## 1 Introduction

Mining data that change over time from fast changing data streams has become a core research problem. Drift detection discovers important distribution changes from labeled classification streams and many drift detectors have been proposed [1,5,8,10]. A drift is signaled when the monitored classification error deviates from its usual value past a certain detection threshold, calculated from a statistical upper bound [6] or a significance technique [9]. The current drift detectors monitor only some form of mean and variance of the classification errors and these errors are used as the only basis for signaling drifts. Currently the detectors do not consider any previous trends in data or drift behaviors. Our proposal incorporates previous drift trends to extend and improve the current drift detection process.

In practice there are many scenarios such as traffic prediction where incorporating previous data trends can improve the accuracy of the prediction process. For example, consider a user using Google Map at home to obtain a fastest route to a specific location. The fastest route given by the system will be based on


A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 417–432, 2015.
DOI: 10.1007/978-3-319-23528-8_26

how congested the roads are at the *current time* (prior to leaving home) but is unable to adapt to situations like upcoming peak hour traffic. The user could be directed to take the main road that is not congested at the time of look up, but may later become congested due to peak hour traffic when the user is *en route*. In this example, combining data such as traffic trends throughout the day can help arrive at a better prediction. Similarly, using historical drift trends, we can derive more knowledge from the stream and when this knowledge is used in the drift detection process, it can improve the accuracy of the predictions.

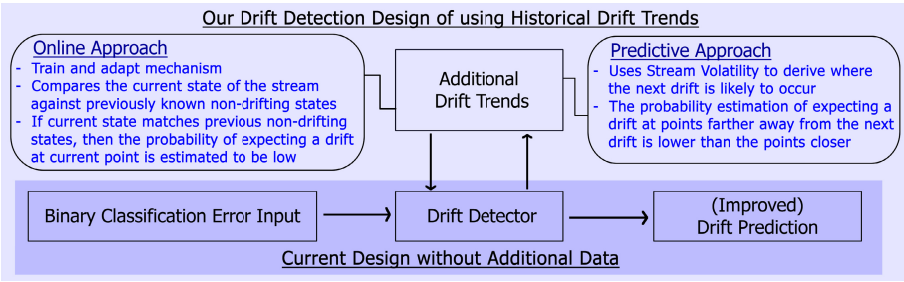


**Fig. 1.** Comparison of current drift detection process v.s. Our proposed design

The main contribution of this paper is the concept of using historical drift trends to estimate the probability of expecting a drift at each point in the stream, which we term the expected drift probability. We propose two approaches to derive this probability: Predictive approach and Online approach. Figure 1 illustrates the comparison of the current drift detection process against our overall proposed design. The Predictive approach uses Stream Volatility [7] to derive a prediction of where the next drift point is likely to occur. Stream Volatility describes the rate of changes in a stream and using the mean of the rate of the changes, we can make a prediction of where the next drift point is. This prediction from Stream Volatility then indicates periods of time where a drift is less likely to be discovered (*e.g.* if the next drift point is predicted to be 100 steps later, then we can assume that drifts are less likely to occur during steps farther away from the prediction). At these times, the Predictive approach will have a low expected drift probability. The predictive approach is suited for applications where the data have some form of cyclic behavior (*i.e.* occurs daily, weekly, etc.) such as the monitoring of oceanic tides, or daily temperature readings for agricultural structures. The Online approach estimates the expected drift probability by first training a model using previous non-drifting data instances. This model represents the state of the stream when drift is not occurring. We then compare how similar the current state of the stream is against the trained model. If the current state matches the model (*i.e.* current state is similar to previous non-drifting states), then we assume that drift is less likely to occur at this current point and derive a low expected drift probability. The Online approach is better suited for fast changing, less predictive applications such as stock market data. We apply the estimated expected drift probability in the state-of-the-art detector ADWIN [1] by adjusting the detection threshold (*i.e.* the statistical upper

bound). When the expected drift probability is low, the detection threshold is adapted and increased to accommodate the estimation. Through experimentation, we offer evidence that using our two new approaches with ADWIN, we achieve a significantly fewer number of false positives.

The paper is structured as follows: in Section 2 we discuss the relevant research. Section 3 details the formal problem definition and preliminaries. In Section 4 our method is presented and we also discuss several key elements and contributions. Section 5 presents our extensive experimental evaluations, and Section 6 concludes the paper.

## 2 Related Work

**Drift Detection:** One way of describing a drift is a statistically significant shift in the distribution of a sample of data which initially represents a single homogeneous distribution to a different data distribution. Gama et al. [4] present a comprehensive survey on drift detection methods and points out that techniques generally fall into four categories: sequential analysis, statistical process control (SPC), monitoring two distributions, and contextual.

The Cumulative Sum [9] and the Page-Hinkley Test [9] are sequential analysis based techniques. They are both memoryless but their accuracy heavily depends on the required parameters, which can be difficult to set. Gama et al. [5] adapted the SPC approach and proposed the Drift Detection Method (DDM), which works best on data streams with sudden drift. DDM monitors the error rate and the variance of the classifying model of the stream.When no changes are detected, DDM works like a lossless learner constantly enlarging the number of stored examples, which can lead to memory problems.

More recently Bifet et al. [1] proposed ADaptive WINdowing (ADWIN) based on monitoring distributions of two subwindows. ADWIN is based on the use of the Hoeffding bound to detect concept change. The ADWIN algorithm was shown to outperform the SPC approach and provides rigorous guarantees on false positive and false negative rates. ADWIN maintains a window ($W$) of instances at a given time and compares the mean difference of any two subwindows ($W_0$ of older instances and $W_1$ of recent instances) from $W$. If the mean difference is statistically significant, then ADWIN removes all instances of $W_0$ considered to represent the old concept and only carries $W_1$ forward to the next test. ADWIN used a variation of exponential histograms and a memory parameter, to limit the number of hypothesis tests.

**Stream Volatility and Volatility Shift:** *Stream Volatility* is a concept introduced in [7], which describes the rate of changes in a stream. A high volatility represents a frequent change in data distribution and a low volatility represents an infrequent change in data distribution. Stream Volatility describes the relationship of proximity between consecutive drift points in the data stream. A *Volatility Shift* is when stream volatility changes (*e.g.* from high volatility to low volatility, or vice versa). Stream volatility is a next level knowledge of detected

changes in data distribution. In the context of this paper, we employ the idea of stream volatility to help derive a prediction of when the next change point will occur in the Predictive approach.

In [7] the authors describe volatility detection as the discovery of a shift in stream volatility. A volatility detector was developed with a particular focus on finding the shift in stream volatility using a relative variance measure. The proposed volatility detector consists of two components: a buffer and a reservoir. The buffer is used to store recent data and the reservoir is used to store an overall representative sample of the stream. A volatility shift is observed when the variance between the buffer and the reservoir is past a significance threshold.

## 3 Preliminaries

Let us frame the problem of drift detection and analysis more formally. Let $S_1 = (x_1, x_2, ..., x_m)$ and $S_2 = (x_{m+1}, ..., x_n)$ with $0 < m < n$ represent two samples of instances from a stream with population means $\mu_1$ and $\mu_2$ respectively. The drift detection problem can be expressed as testing the null hypothesis $H_0$ that $\mu_1 = \mu_2$, *i.e.* the two samples are drawn from the same distribution against the alternate hypothesis $H_1$ that they are drawn from different distributions with $\mu_1 \neq \mu_2$. In practice the underlying data distribution is unknown and a test statistic based on sample means is constructed by the drift detector. If the null hypothesis is accepted incorrectly when a change has occurred then a false negative has occurred. On the other hand if the drift detector accepts $H_1$ when no change has occurred in the data distribution then a false positive has occurred. Since the population mean of the underlying distribution is unknown, sample means need to be used to perform the above hypothesis tests. The hypothesis tests can be restated as the following. We accept hypothesis $H_1$ whenever $Pr(|\hat{\mu_1} - \hat{\mu_2}|) \geq \epsilon) \leq \delta$, where the parameter $\delta \in (0, 1)$ and controls the maximum allowable false positive rate, while $\epsilon$ is the test statistic used to model the difference between the sample means and is a function of $\delta$.

## 4 Our Concept and Design

We present how to use historical drift trend to estimate the probability of expecting a drift at every point in the stream using our Predictive approach in Section 4.1 and Online approach in Section 4.2. In Section 4.3 we describe how the expected drift probability is applied onto drift detector ADWIN.

### 4.1 Predictive Approach

The Predictive approach is based on Stream Volatility. Recall that Stream Volatility describes the rate of changes in the stream. The mean volatility value is the average interval between drift points, denoted $\mu_{volatility}$, and is derived from the history of drift points in the stream. For example, a stream with drift

points at times $t = 50$, $t = 100$, and $t = 150$ will have a mean volatility value of 50. The $\mu_{volatility}$ is then used to provide an estimate of a relative position of where the next drift point, denoted $t_{drift.next}$, is likely to occur. In other words, $t_{drift.next} = t_{drift.previous} + \mu_{volatility}$, where $t_{drift.previous}$ is the location of the previous signaled drift point and $t_{drift.previous} < t_{drift.next}$.

The expected drift probability at points $t_x$ in the stream is denoted as $\phi_{t_x}$. We use the $t_{drift.next}$ prediction to derive $\phi_{t_x}$ at each time point $t_x$ in the stream between the previous drift and the next predicted drift point, $t_{drift.previous} < t_x < t_{drift.next}$. When $t_x$ is distant from $t_{drift.next}$, the probability $\phi_{t_x}$ is smaller and as $t_x$ progresses closer to $t_{drift.next}$, $\phi_{t_x}$ is progressively increased.

We propose two variations of deriving $\phi_{t_x}$ based on next drift prediction: one based on the sine function and the other based on sigmoid function. Intuitively the sine function will assume that the midpoint of previous drift and next drift is the least likely point to observe a drift whereas the sigmoid function will assume that immediately after a drift, the probability of observing a drift is low until the stream approaches the next drift prediction.

The calculation of $\phi_{t_x}$ using the sine and the sigmoid functions at a point $t_x$ where $t_{drift.previous} < t_x < t_{drift.next}$ are defined as:

$$\phi_{t_x}^{\sin} = 1 - \sin(\pi \cdot t_r) \text{ and } \phi_{t_x}^{sigmoid} = 1 - \frac{1 - t_r}{0.01 + |1 - t_r|}$$

where $t_r = (t_x - t_{drift.previous})/(t_{drift.next} - t_{drift.previous})$

The Predictive approach is applicable when the volatility of the stream is relatively stable. When the volatility of the stream is unstable and highly variable, the Predictive approach will be less reliable at predicting where the next drift point is likely to occur. When this situation arises, the Online approach (described in Section 4.2) should be used.

### 4.2 Online Adaptation Approach

The Online approach estimates the expected drift probability by first training a model using previous non-drifting data. This model represents the state of the stream when drift is not occurring. We then compare how similar the current state of the stream is against the trained model. If the current state matches the model (*i.e.* current state is similar to previous non-drifting states), then we assume that drift is less likely to occur at this current point and derive a low expected drift probability. Unlike the Predictive approach, which predicts where a drift point might occur, the Online approach approximates the unlikelihood of a drift occurring by comparing the current state of the stream against the trained model representing non-drifting states.

The Online approach uses a sliding block $B$ of size $b$ that keeps the most recent value of binary inputs. The mean of the binary contents in the block at time $t_x$ is given as $\mu_{B_{tx}}$ where the block $B$ contains samples with values $v_{x-b}, \cdots, v_x$ of the transactions $t_{x-b}, \cdots, t_x$. The simple moving average of the previous $n$ inputs is also maintained where: $(v_{x-b-n} + \cdots + v_{x-b})/n$. The state

of the stream at any given time $t_x$ is represented using the *Running Magnitude Difference*, denoted as $\gamma$, and given by: $\gamma_{t_x} = \mu_{B_{tx}} - \text{MovingAverage}$.

We collect a set of $\gamma$ values $\gamma_1, \gamma_2, \cdots, \gamma_x$ using previous non-drifting data to build a Gaussian training model. The Gaussian model will have the mean $\mu_\gamma$ and the variance $\sigma_\gamma$ of the training set of $\gamma$ values. The set of $\gamma$ values reflect the stability of the mean of the binary inputs. A stream in its non-drifting states will have a set of $\gamma$ values that tend to a mean of 0.

To calculate the expected drift probability in the running stream at a point $t_x$, the estimation $\phi_{t_x}$ is derived by comparing the Running Magnitude Difference $\gamma_{t_x}$ against the trained Gaussian model with $\alpha$ as the threshold, the probability calculation is given as:

$$\phi_{t_x}^{online} = 1 \text{ if } f(\gamma_{t_x}, \mu_\gamma, \sigma_\gamma) \leq \alpha$$

and

$$\phi_{t_x}^{online} = 0 \text{ if } f(\gamma_{t_x}, \mu_\gamma, \sigma_\gamma) > \alpha$$

where

$$f(\gamma_{t_x}, \mu_\gamma, \sigma_\gamma) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\gamma_{t_x} - \mu_\gamma)^2}{2{\sigma_\gamma}^2}}$$

For example, using previous non-drifting data we build a Gaussian model with $\mu_\gamma = 0.0$ and $\sigma_\gamma = 0.05$, if $\alpha = 0.1$ and we observe that the current state $\gamma_{t_x} = 0.1$, then the expected drift probability is 1 because $f(0.1, 0, 0.05) = 0.054 \leq \alpha$.

In a stream environment, the *i.i.d.* property of incoming random variables in the stream is generally assumed. Although at first glance it may appear that a trained Gaussian model is not suitable to be used in this setting, the central limit theorem provides justification. In drift detection a drift primarily refers to the *real concept drift*, which is a change in the posterior distributions of data $p(y|X)$, where $X$ is the set of attributes and $y$ is the target class label. When the distribution of $X$ changes, the class $y$ might also change affecting the predictive power of the classifier and signals a a drift. Since the Gaussian model is trained using non-drifting data, we assume that the collected $\gamma$ value originates from the same underlying distribution and remains stable in the non-drifting set of data. Although the underlying distribution of the set of $\gamma$ values is unknown, the Central Limit Theorem justifies that the mean of a sufficiently large number of random samples will be approximately normally distributed regardless of the underlying distribution. Thus, we can effectively approximate the set of $\gamma$ values with a Gaussian model. To confirm that the Central Limit Theorem is valid in our scenario, we have generated sets of non-drifting supervised two class labeled streams using the rotating hyperplane generator with the set of numeric attributes $X$ generated from different source distributions such as uniform, binomial, exponential, and Poisson. The sets of data are then run through a Hoeffding Tree Classifier to obtain the binary classification error inputs and the set of $\gamma$ values are gathered using the Online approach. We plot each set of the $\gamma$ values and demonstrate that the distribution indeed tends to Gaussian as justified by the Central Limit Theorem in Figure 2.

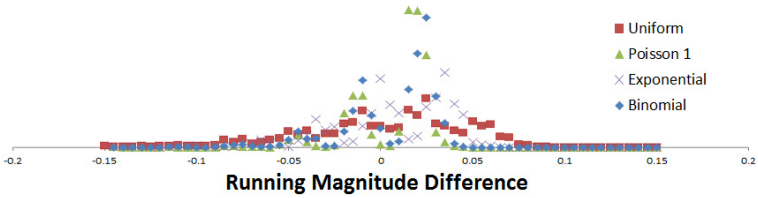


**Fig. 2.** Demonstration of Central Limit Theorem

### 4.3 Application onto ADWIN

In Sections 4.1 and 4.2 we have described two approaches at calculating the expected drift probability $\phi$ and in this section we show how to apply the discovered $\phi$ in the detection process of ADWIN.

ADWIN relies on using the Hoeffding bound with Bonferroni correction [1] as the detection threshold. The Hoeffding bound provides guarantee that a drift is signalled with probability at most $\delta$ (a user defined parameter): $Pr(|\mu_{W_0} - \mu_{W_1}| \geq \epsilon) \leq \delta$ where $\mu_{W_0}$ is the sample mean of the reference window of data, $W_0$, and $\mu_{W_1}$ is the sample mean of the current window of data, $W_1$. The $\epsilon$ value is a function of $\delta$ parameter and is the test statistic used to model the difference between the sample mean of the two windows. Essentially when the difference in sample means between the two windows is greater than the test statistic $\epsilon$, a drift will be signaled. $\epsilon$ is given by the Hoeffding bound with Bonferroni correction as:

$$\epsilon_{hoeffding} = \sqrt{\frac{2}{m} \cdot \sigma_W^2 \cdot ln\frac{2}{\delta'}} + \frac{2}{3m}\ ln\frac{2}{\delta'}$$

where $m = \frac{1}{1/|W_0|+1/|W_1|}$, $\delta' = \frac{\delta}{|W_0|+|W_1|}$.

We incorporate $\phi$, expected drift probability, onto ADWIN and propose an Adaptive bound which adjusts the detection threshold of ADWIN in reaction to the probability of seeing a drift at different time $t_x$ across the stream. When $\phi$ is low, the Adaptive bound (detection threshold) is increased to accommodate the estimation that drift is less likely to occur.

The $\epsilon$ of the Adaptive bound is derived as follows:

$$\epsilon_{adaptive} = (1 + \beta \cdot (1 - \phi))\sqrt{\frac{2}{m} \cdot \sigma_W^2 \cdot ln\frac{2}{\delta'}} + \frac{2}{3m}\ ln\frac{2}{\delta'}$$

where $\beta$ is a tension parameter that controls the maximum allowable adjustment, usually set below 0.5. A comparison of the Adaptive bound using the Predictive approach versus the original Hoeffding bound with Bonferroni correction is shown in Figure 3. In such cases, we can see that by using the Adaptive bound derived from the Predictive approach, we reduce the number of false positives that would have otherwise been signaled by Hoeffding bound.

The Adaptive bound is based on adjusting the Hoeffding bound and maintains similar statistical guarantees as the original Hoeffding bound. We know

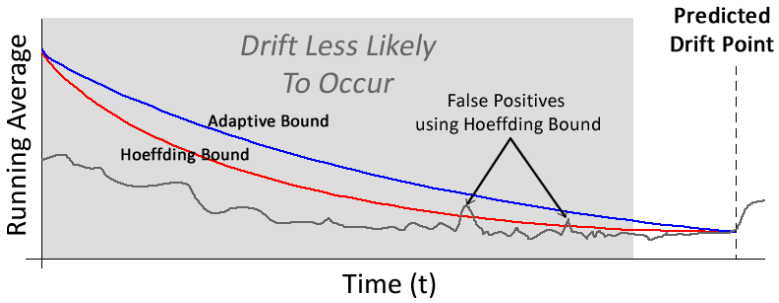


**Fig. 3.** Demonstration of Adaptive bound vs. Hoeffding bound

that the Hoeffding bound provides guarantee that a drift is signaled with probability at most $\delta$:

$$Pr(|\mu_{W_0} - \mu_{W_1}| \geq \epsilon) \leq \delta$$

and since

$$\epsilon_{adaptive} \geq \epsilon_{hoeffding}$$

therefore,

$$Pr(|\mu_{W_0} - \mu_{W_1}| \geq \epsilon_{adaptive}) \leq Pr(|\mu_{W_0} - \mu_{W_1}| \geq \epsilon_{hoeffding}) \leq \delta$$

As a result, we know that the Adaptive bound is *at least as confident* as the Hoeffding bound and offer the same guarantees as the Hoeffding bound given $\delta$.

The Predictive approach should be used when the input stream allows for an accurate prediction of where the next drift point is likely to occur. A good prediction of next drift point will show major performance increases. The Predictive approach has tolerance to incorrect next drift predictions to a certain margin of error. However, when the stream is too volatile and fast changing to the extent where using volatility to predict the next drift point is unreasonable, the Online approach should be used. The benefits of using the Online approach is that the performance of the approach is not affected irrespective of whether the stream is volatile or not. The Online approach is also better suited for scenarios where a good representative non-drifting training data can be provided.

When using the Predictive approach to estimate $\phi$ in the Adaptive bound, we note that an extra *warning level* mechanism should be added. The Predictive approach uses the mean volatility value, the average number of transactions between each consecutive drift points in the past, to derive where the next drift point is likely to occur. The mean volatility value is calculated based on previous detection results of the drift detector. The Adaptive bound with the Predictive approach works by increasing the detection threshold at points before the prediction. In cases where the true drift point arrives before the prediction, the drift detector will experience a higher detection delay due to the higher detection threshold of the Adaptive bound. This may affect future predictions based on the mean volatility value. Therefore, we note the addition of the *warning level*, which is when the Hoeffding bound is passed but not the Adaptive bound. A real

drift will pass the Hoeffding bound, then pass the Adaptive bound, while a false positive might pass Hoeffding bound but not the Adaptive bound. When a drift is signaled by the Adaptive bound, the mean volatility value is updated using the drift point found when Hoeffding bound was first passed. The addition of the Hoeffding as the warning level resolves the issue that drift points found by the Adaptive bound might influence future volatility predictions.

The $\beta$ value is a tension parameter used to control the degree at which the statistical bound is adjusted based on drift expectation probability estimation $\phi$. Setting a higher $\beta$ value will increase the magnitude of adjustment of the bound. One sensible way to set the $\beta$ parameter is to base its value on the confidence of the $\phi$ estimation. If the user is confident in the $\phi$ estimation, then setting a high $\beta$ value ($e.g.$ 0.5) will significantly reduce the number of false positives while still detecting all the real drifts. If the user is less confident in the $\phi$ estimation, then $\beta$ can be set low ($e.g.$ 0.1) to make sure drifts of significant value are picked up. An approach for determining the $\beta$ value is: $\beta = Pr(a) - Pr(e)/2 \cdot (1 - Pr(e))$ where $Pr(a)$ is the confidence of the $\phi$ estimation and $Pr(e)$ is the confidence of estimation by chance. In most instances $Pr(e)$ should be set at 0.6 as any estimation with confidence lower than 0.6 we can consider as a poor estimation. In practice by setting $\beta$ at 0.1 reduces the number of false positives found by 50-70% when incorporating our design into ADWIN while maintaining similar true positive rates and detection delay compared to without using our design.

## 5 Experimental Evaluation

In this section we present the experimental evaluation of applying our expected drift probability $\phi$ onto ADWIN with the Adaptive bound. We test with both the Predictive approach and the Online approach. Our algorithms are coded in Java and all of the experiments are run on an Intel Core i5-2400 CPU @ 3.10 GHz with 8GB of RAM running Windows 7.

We evaluate with different sets of standard experiments in drift detection: false positive test, true positive test, and false negative test using various $\beta$ (tension parameter) values from 0.1 to 0.5. The detection delay is also a major indicator of performance therefore reported and compared in the experiments. Our proposed algorithms have a run-time of $< 2ms$. Further supplementary materials and codes can be found online

### 5.1 False Positive Test

In this test we compare the false positives found between using only ADWIN detector against using our Predictive and Online approaches on top of ADWIN. For this test we replicate the standard false positive test used in [1]. A stream of 100,000 bits with no drifts generated from a Bernoulli distribution with $\mu = 0.2$ is used. We vary the $\delta$ parameter and the $\beta$ tension parameter and run 100 iterations for all experiments. The Online approach is run with a 10-fold cross validation. We use $\alpha = 0.1$ for the Online approach.

**Table 1.** False Positive Rate Comparison

| **Predictive Approach - Sine Function** | | | | | | |
|---|---|---|---|---|---|---|
| | | Adaptive Bound | | | | |
| $\delta$ | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.05 | 0.0014 | 0.0006 | 0.0003 | 0.0002 | 0.0001 | 0.0001 |
| 0.1 | 0.0031 | 0.0015 | 0.0008 | 0.0005 | 0.0003 | 0.0002 |
| 0.3 | 0.0129 | 0.0072 | 0.0042 | 0.0027 | 0.0019 | 0.0015 |
| **Predictive Approach - Sigmoid Function** | | | | | | |
| | | Adaptive Bound | | | | |
| $\delta$ | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.05 | 0.0014 | 0.0004 | 0.0002 | 0.0001 | 0.0001 | 0.0001 |
| 0.1 | 0.0031 | 0.0011 | 0.0004 | 0.0003 | 0.0002 | 0.0002 |
| 0.3 | 0.0129 | 0.0057 | 0.0030 | 0.0018 | 0.0015 | 0.0014 |
| **Online Approach** | | | | | | |
| $\delta$ | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.05 | 0.0014 | 0.0006 | 0.0005 | 0.0005 | 0.0005 | 0.0005 |
| 0.1 | 0.0031 | 0.0014 | 0.0012 | 0.0011 | 0.0011 | 0.0011 |
| 0.3 | 0.0129 | 0.0073 | 0.0063 | 0.0059 | 0.0058 | 0.0058 |

The results are shown in Table 1 and we observe that both the sine function and sigmoid function with the Predictive approach are effective at reducing the number of false positives in the stream. In the best case scenario the number of false positive was reduced by 93%. Even with a small $\beta$ value of 0.1, we still observe an approximately 50-70% reduction in the number of false positives. For the Online approach we observe around a 65% reduction.

### 5.2 True Positive Test

In the true positive test we test the accuracy of the three different setting at detecting true drift points. In addition, we look at the detection delay associated with the detection of the true positives. For this test we replicate the true positive test used in [1]. Each stream consists of 1 drift at different points of volatility with varying magnitudes of drift and drift is induced with different slope values over a period of 2000 steps. For each set of parameter values, the experiments are run over 100 iterations using ADWIN as the drift detector with $\delta = 0.05$. The Online approach was run with a 10-fold cross validation.

We observed that for all slopes, the true positive rate for using all different settings (ADWIN only, Predictive_Sine, and Predictive_Sigmoid, Online) is 100%. There was no notable difference between using ADWIN only, Predictive

**Table 2.** True Positive Test: Detection Delay

| Predictive Approach - Sigmoid Function | | | | | | |
|---|---|---|---|---|---|---|
| Slope | Hoeffding | Adaptive Bound | | | | |
| | | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.0001 | 882±(181) | 882±(181) | 880±(181) | 874±(191) | 874±(191) | 874±(191) |
| 0.0002 | 571±(113) | 569±(112) | 564±(114) | 562±(119) | 562±(119) | 562±(119) |
| 0.0003 | 441±(83) | 440±(82) | 438±(82) | 436±(86) | 436±(86) | 436±(86) |
| 0.0004 | 377±(71) | 375±(68) | 373±(69) | 371±(72) | 371±(72) | 371±(72) |
| **Online Approach** | | | | | | |
| Slope | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.0001 | 882±(181) | 941±(180) | 975±(187) | 1001±(200) | 1015±(211) | 1033±(220) |
| 0.0002 | 571±(113) | 597±(116) | 611±(123) | 620±(130) | 629±(136) | 632±(140) |
| 0.0003 | 441±(83) | 460±(90) | 469±(94) | 472±(96) | 475±(100) | 476±(101) |
| 0.0004 | 377±(71) | 389±(73) | 394±(74) | 398±(79) | 398±(79) | 399±(80) |

approach, and Online approach in terms of accuracy. The results for the associated detection delay on gradual drift stream are shown in Table 2. We note that the Sine and Sigmoid functions yielded similar results and we only present one of them here. We see that the detection delays remain stable between ADWIN only and the Predictive approach as this test assumes an accurate next drift prediction from volatility and does not have any significant variations. The Online approach observed a slightly higher delay due to the nature of the approach (within one standard deviation of Hoeffding bound delay). There is an increase in delay when $\beta$ was varied only in the Online approach.

### 5.3 False Negative Test

The false negative test experiments if predictions are correct. Hence, the experiments are carried out on the Predictive approach and not the Online approach.

For this experiment we generate streams with 100,000 bits containing exactly one drift at a pre-specified location before the presumed drift location (100,000). We experiment with 3 locations at steps 25,000 (the 1/4 point), 50,000 (the 1/2 point), and 75,000 (the 3/4 point). The streams are generated with different drift slopes modelling both gradual and abrupt drift types. We feed the Predictive approach a drift prediction at 100,000. We use ADWIN with $\delta = 0.05$.

In Table 3 we show the detection delay results for varying $\beta$ and drift types/slopes when the drift is located at the 1/4 point. We observe from the table that as we increase the drift slope, the delay decreases. This is because a drift of a larger magnitude is easier to detect and thus found faster. As we increase $\beta$ we can see a positive correlation with delay. This is an apparent tradeoff with adapting to a tougher bound. In most cases the increase in delay associated with an unpredicted drift is still acceptable taking into account the

**Table 3.** Delay Comparison: 1/4 drift point

| **Sine Function** | | | | | | |
|---|---|---|---|---|---|---|
| | | Adaptive Bound | | | | |
| Slope | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.4 | 107±(37) | 116±(39) | 129±(42) | 139±(43) | 154±(47) | 167±(51) |
| 0.6 | 52±(12) | 54±(11) | 57±(11) | 61±(12) | 64±(14) | 67±(15) |
| 0.8 | 27±(10) | 28±(11) | 32±(14) | 39±(16) | 44±(15) | 50±(12) |
| 0.0001 | 869±(203) | 923±(193) | 972±(195) | 1026±(200) | 1090±(202) | 1151±(211) |
| 0.0002 | 556±(121) | 593±(117) | 634±(106) | 664±(109) | 692±(116) | 727±(105) |
| 0.0003 | 434±(89) | 463±(91) | 488±(84) | 514±(83) | 531±(80) | 557±(75) |
| 0.0004 | 367±(71) | 384±(76) | 403±(73) | 420±(70) | 439±(71) | 457±(69) |
| **Sigmoid Function** | | | | | | |
| | | Adaptive Bound | | | | |
| Slope | Hoeffding | $\beta = 0.1$ | $\beta = 0.2$ | $\beta = 0.3$ | $\beta = 0.4$ | $\beta = 0.5$ |
| 0.4 | 108±(37) | 121±(40) | 136±(42) | 156±(46) | 176±(51) | 196±(56) |
| 0.6 | 52±(12) | 54±(12) | 57±(11) | 61±(12) | 64±(14) | 67±(15) |
| 0.8 | 27±(10) | 28±(11) | 32±(14) | 39±(16) | 44±(15) | 50±(12) |
| 0.0001 | 869±(203) | 937±(200) | 1013±(198) | 1091±(201) | 1177±(216) | 1259±(221) |
| 0.0002 | 556±(121) | 605±(110) | 657±(110) | 695±(115) | 738±(103) | 776±(102) |
| 0.0003 | 434±(89) | 474±(87) | 508±(87) | 535±(78) | 567±(76) | 593±(76) |
| 0.0004 | 367±(71) | 390±(75) | 415±(71) | 442±(70) | 471±(65) | 492±(61) |

**Table 4.** Delay Comparison: Varying drift point

| **Gradual Drift 0.0001** | | | | | | |
|---|---|---|---|---|---|---|
| | Sine | | | Sigmoid | | |
| $\beta$ | 1/4 point | 1/2 point | 3/4 point | 1/4 point | 1/2 point | 3/4 point |
| Hoeffding | 869±(203) | 885±(183) | 872±(183) | 869±(203) | 885±(183) | 872±(183) |
| 0.1 | 923±(192) | 956±(187) | 930±(178) | 937±(200) | 955±(185) | 948±(176) |
| 0.2 | 972±(195) | 1026±(197) | 982±(173) | 1013±(198) | 1022±(195) | 1020±(183) |
| 0.3 | 1026±(200) | 1095±(203) | 1029±(182) | 1091±(201) | 1091±(202) | 1096±(197) |
| 0.4 | 1090±(202) | 1182±(212) | 1087±(192) | 1177±(216) | 1176±(207) | 1167±(204) |
| 0.5 | 1151±(211) | 1253±(217) | 1133±(198) | 1259±(221) | 1243±(215) | 1245±(214) |

magnitude of false positive reductions and the assumption that unexpected drifts should be less likely to occur when volatility predictions are relatively accurate.

Table 4 compares the delay when the drift is thrown at different points during the stream. It can be seen that the sine function does have a slightly higher delay when the drift is at the 1/2 point. This can be traced back to the sine function where the mid-point is the peak of the offset. In general the variations are within reasonable variation and the differences are not significant.

### 5.4 Real-World Data: Power Supply Dataset

This dataset is obtained from the Stream Data Mining Repository[1]. It contains the hourly power supply from an Italian electricity company from two sources. The measurements from the sources form the attributes of the data and the class label is the hour of the day from which the measurement is taken. The drifts in this dataset are primarily the differences in power usage between different seasons where the hours of daylight vary. We note that because real-world datasets do not have ground truths for drift points, we are unable to report true positive rates, false positive rates, and detection delay. The main objective is to compare the behavior of ADWIN only versus our designs on real-world data and show that they do find drifts at similar locations.

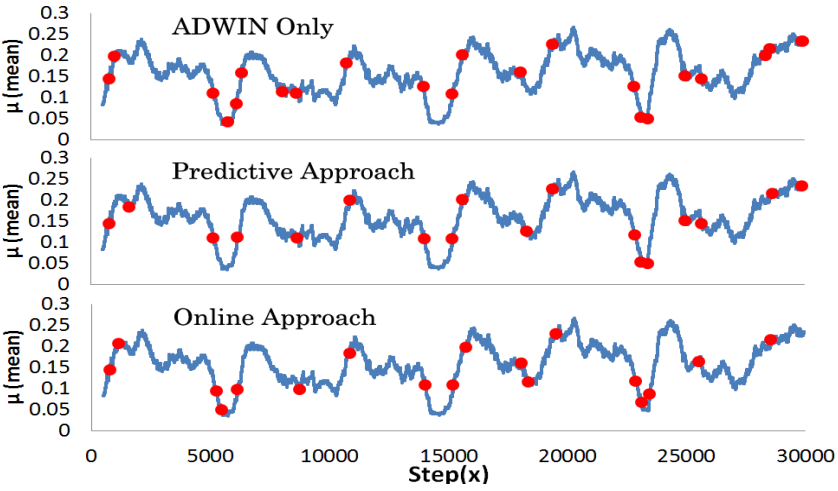


**Fig. 4.** Power Supply Dataset (drift points are shown with red dots)

Figure 4 shows the comparison using the drifts found between ADWIN Only and our Predictive and Online approaches. We observe that the drift points are fired at similar locations to the ADWIN only approach.

### 5.5 Case Study: Incremental Classifier

In this study we apply our design into incremental classifiers in data streams. We experiment with both synthetically generated data and real-world datasets. First, we generate synthetic data with the commonly used SEA concepts generator introduced in [11]. Second, we use Forest Covertype and Airlines real-world datasets from MOA website[2]. Note that the ground truth for drifts is not available for the real-world datasets. In all of the experiments we run VFDT [3], an

[1] http://www.cse.fau.edu/~xqzhu/stream.html
[2] moa.cms.waikato.ac.nz/datasets/

incremental tree classifier, and compare the prediction accuracy and learning time of the tree using three settings: VFDT without drift detection, VFDT with ADWIN, and VFDT with our approaches and Adaptive bound. In the no drift detection setting, the tree learns throughout the entire stream. In the other settings, the tree is rebuilt using the next $n$ instances when a drift is detected. For SEA data $n = 10000$ and for real-world $n = 300$. We used a smaller $n$ for the real-world datasets because they contain fewer instances and more drift points.

The synthetic data stream is generated from MOA [2] using the SEA concepts with 3 drifts evenly distributed at 250k, 500k, and 750k in a 1M stream. Each section of the stream is generated from one of the four SEA concept functions. We use $\delta = 0.05$ for ADWIN, Sine function for Predictive approach and $\beta = 0.5$ for both approaches and $\alpha = 0.1$ for Online approach. Each setting is run over 30 iterations and the observed results are shown in Table 5.

The results show that by using ADWIN only, the overall accuracy of the classifier is improved. There is also a reduction in the learning time because only parts of the stream are used for learning the classifier as opposed to no drift detection where the full stream is used. Using our Predictive approach and the Online approach showed a further reduction in learning time and an improvement in accuracy. An important observation is the reduction in the number of drifts detected in the stream and an example of drift points is shown in Table 6. We discovered that using Predictive and Online approaches found less false positives.

**Table 5.** Incremental Classifier Performance Comparisons

| **SEA Concept Generator (3 actual drifts)** | | | |
|---|---|---|---|
| Setting | Learning Time (ms) | Accuracy | Drifts Detected |
| No Drift Detection | 2763.67±(347.34) | 85.71±(0.06)% | - |
| ADWIN Only | 279.07±(65.06) | 87.36±(0.15)% | 13.67±(3.66) |
| Predictive Approach | 178.63±(41.06) | 87.44±(0.22)% | 8.67±(1.72) |
| Online Approach | **161.10±(38.93)** | **87.49±(0.22)%** | **6.60±(1.56)** |

| **Real-World Dataset: Forest Covertype** | | | |
|---|---|---|---|
| Setting | Learning Time (ms) | Accuracy | Drifts Detected |
| No Drift Detection | 44918.47±(149.44) | 83.13% | - |
| ADWIN Only | 45474.57±(226.40) | **89.37%** | 1719 |
| Predictive Approach | 45710.87±(226.40) | 89.30% | 1701 |
| Online Approach | 45143.07±(212.40) | 89.09% | **1602** |

| **Real-World Dataset: Airlines** | | | |
|---|---|---|---|
| Setting | Learning Time (ms) | Accuracy | Drifts Detected |
| No Drift Detection | 2051.87±(141.42) | 67.44% | - |
| ADWIN Only | **1654.37±(134.79)** | **75.96%** | 396 |
| Predictive Approach | **1602.07±(120.24)** | 75.80% | 352 |
| Online Approach | **1637.27±(124.16)** | 75.79% | **320** |

**Table 6.** SEA Dataset Drift Point Comparison

| SEA Sample: induced drifts at 250k, 500k, and 750k (false positives colored) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADWIN | 19167 | 102463 | 106367 | 250399 | 407807 | 413535 | 432415 | 483519 | 489407 | 500223 | 739423 | 750143 |
| Predict. | 19167 | 102463 | 106399 | 250367 | | | | | | 500255 | | 750239 |
| Online | 19167 | | | 251327 | | | | | | 500255 | | 750143 |

The reduction in the number of drifts detected means that the user does not need to react to unnecessary drift signals. In the real-world dataset experiments, we generally observe a similar trend to the synthetic experiments. Overall the classifier's accuracy is improved when our approaches are applied. Using ADWIN only yields the highest accuracy, however, it is only marginally higher than our approaches while using our approaches the number of drifts detected is reduced. With real-world datasets, we unfortunately do not have the ground truths and cannot produce variance in the accuracy and number of drifts detected, but the eliminated drifts using our approaches did not have apparent effects on the accuracy of the classifier and thus are more likely to be false positives or less significant drifts. Although the accuracy results are not statistically worse or better, we observe a reduction in the number of drifts detected. In scenarios where drift signals incur high costs of action, having a lower number of detected drifts while maintaining similar accuracy is in general more favorable.

## 6 Conclusion and Future Work

We have described a novel concept of estimating the probability of expecting a drift at each point in the stream based on historical drift trends such as Stream Volatility. To the best of our knowledge this work is the first that investigate this idea. We proposed two approaches to derive the expected drift probability: Predictive approach and Online approach. The Predictive approach uses Stream Volatility [7] to derive a prediction of where the next drift point is likely to occur and based on that prediction the expected drift probability is determined using the proximity of the points to the next drift prediction. The Online approach estimates the expected drift probability by first training a model using previous non-drifting data instances and compare the current state of the stream against the trained model. If the current state matches the model then we assume that drift is less likely to occur at this current point and derive a low expected drift probability. We incorporate the derived expected drift probability in the state-of-the-art detector ADWIN by adjusting the statistical upper bound. When the expected drift probability is low, the bound is increased to accommodate the estimation. Through experimentation, we offer evidence that using our design in ADWIN, we can achieve significantly fewer number of false positives.

Our future work includes applying the Adaptive bound onto other drift detection techniques that utilize similar statistical upper bounds such as SEED [7]. We also want to look at using other stream characteristics such as the types of drifts (*e.g.* gradual and abrupt) to derive the expected drift probability.

## References

1. Bifet, A., Gavaldá, R.: Learning from time-changing data with adaptive windowing. In: SIAM International Conference on Data Mining (2007)
2. Bifet, A., Holmes, G., Pfahringer, B., Read, J., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: MOA: a real-time analytics open source framework. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 617–620. Springer, Heidelberg (2011)
3. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 71–80 (2000)
4. Gama, J.A., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Computing Surveys **46**(4), 44:1–44:37 (2014)
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
6. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association **58**, 13–29 (1963)
7. Huang, D.T.J., Koh, Y.S., Dobbie, G., Pears, R.: Detecting volatility shift in data streams. In: 2014 IEEE International Conference on Data Mining (ICDM), pp. 863–868 (2014)
8. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the 30th International Conference on VLDB, pp. 180–191. VLDB Endowment (2004)
9. Page, E.: Continuous inspection schemes. Biometrika, 100–115 (1954)
10. Pears, R., Sakthithasan, S., Koh, Y.S.: Detecting concept change in dynamic data streams - A sequential approach based on reservoir sampling. Machine Learning **97**(3), 259–293 (2014)
11. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001, pp. 377–382. ACM, New York (2001)