

Article

An Experimental Analysis of Drift Detection Methods on Multi-Class Imbalanced Data Streams

Abdul Sattar Palli ^{1,2,*}, Jafreezal Jaafar ^{1,3}, Heitor Murilo Gomes ^{4,5}, Manzoor Ahmed Hashmani ^{1,6}  and Abdul Rehman Gilal ¹

¹ Department of Computer and Information Sciences, Universiti Teknologi PETRONAS (UTP), Tronoh 32610, Perak, Malaysia

² Anti Narcotics Force, Ministry of Narcotics Control, Rawalpindi 46000, Pakistan

³ Centre for Research in Data Science, Universiti Teknologi PETRONAS (UTP), Tronoh 32610, Perak, Malaysia

⁴ School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand

⁵ AI Institute, University of Waikato, Wellington 3240, New Zealand

⁶ High-Performance Cloud Computing Centre (HPC3), Universiti Teknologi PETRONAS (UTP), Tronoh 32610, Perak, Malaysia

* Correspondence: abdul_20000263@utp.edu.my

Featured Application: The industrial sensor-based application generates continuous non-stationary data streams which change over time. By analyzing the performance of existing change detection methods, the selection of the best performing method can be achieved for application in an industrial environment to early detect the fault or unusual change and to reduce the maintenance cost.



Citation: Palli, A.S.; Jaafar, J.; Gomes, H.M.; Hashmani, M.A.; Gilal, A.R. An Experimental Analysis of Drift Detection Methods on Multi-Class Imbalanced Data Streams. *Appl. Sci.* **2022**, *12*, 11688. <https://doi.org/10.3390/app122211688>

Academic Editors: Jiaqi Li, Božidar Šarler, Haiping Liu and Jian Zhang

Received: 15 October 2022

Accepted: 12 November 2022

Published: 17 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: The performance of machine learning models diminishes while predicting the Remaining Useful Life (RUL) of the equipment or fault prediction due to the issue of concept drift. This issue is aggravated when the problem setting comprises multi-class imbalanced data. The existing drift detection methods are designed to detect certain drifts in specific scenarios. For example, the drift detector designed for binary class data may not produce satisfactory results for applications that generate multi-class data. Similarly, the drift detection method designed for the detection of sudden drift may struggle with detecting incremental drift. Therefore, in this experimental investigation, we seek to investigate the performance of the existing drift detection methods on multi-class imbalanced data streams with different drift types. For this reason, this study simulated the streams with various forms of concept drift and the multi-class imbalance problem to test the existing drift detection methods. The findings of current study will aid in the selection of drift detection methods for use in developing solutions for real-time industrial applications that encounter similar issues. The results revealed that among the compared methods, DDM produced the best average F1 score. The results also indicate that the multi-class imbalance causes the false alarm rate to increase for most of the drift detection methods.

Keywords: fault detection; concept drift; drift detection; class imbalance; multi-class classification; data stream mining

1. Introduction

The difficulty of learning from streaming data with concept drift has increased attention in the area of online learning in recent years since this phenomenon happens in many real-time applications like predicting RUL [1–3] fault detection [4–9], and risk management [10–14]. These applications generate continuous non-stationary data streams. The non-stationary nature of the stream causes a change in the statistical properties of data which leads to concept drift and class imbalance problems [15,16]. The concept drift includes virtual drift, real drift, or hybrid drift [17]. Let us assume, at time t , we have x_t as

an input vector for the Machine Learning (ML) model and y_t as the corresponding target output vector. The virtual drift occurs when the distribution of input data $p_t(x)$ changes over time, but the posterior probability of the output $p_t(y|x)$, which shows the mapping relationship between x_t and y_t , does not change with time. In simple words, the data, even after facing changes in input features, still represent the same target output. In real drifts, the posterior probability distributions $p_t(y|x)$ change over time, but this change is not caused by changes in $p_t(x)$. In contrast, hybrid drift is the combination of both virtual drift and real drift. These major types of drift are further classified into different types based on the frequency of underlying change with respect to time, which are sudden, incremental, gradual, and reoccurring [17,18]. Figure 1 illustrates all four types of concept drift. The basic definition of each of the drift type is as follows:

Sudden: The concept of data abruptly changes from one concept to another

Incremental: The concept of data incrementally changes with a constant speed from one concept to another

Gradual: Similar to incremental drift but speed of change of one concept to another is not constant

Reoccurring: The reoccurring drift disappears, and after a certain time period it reappears

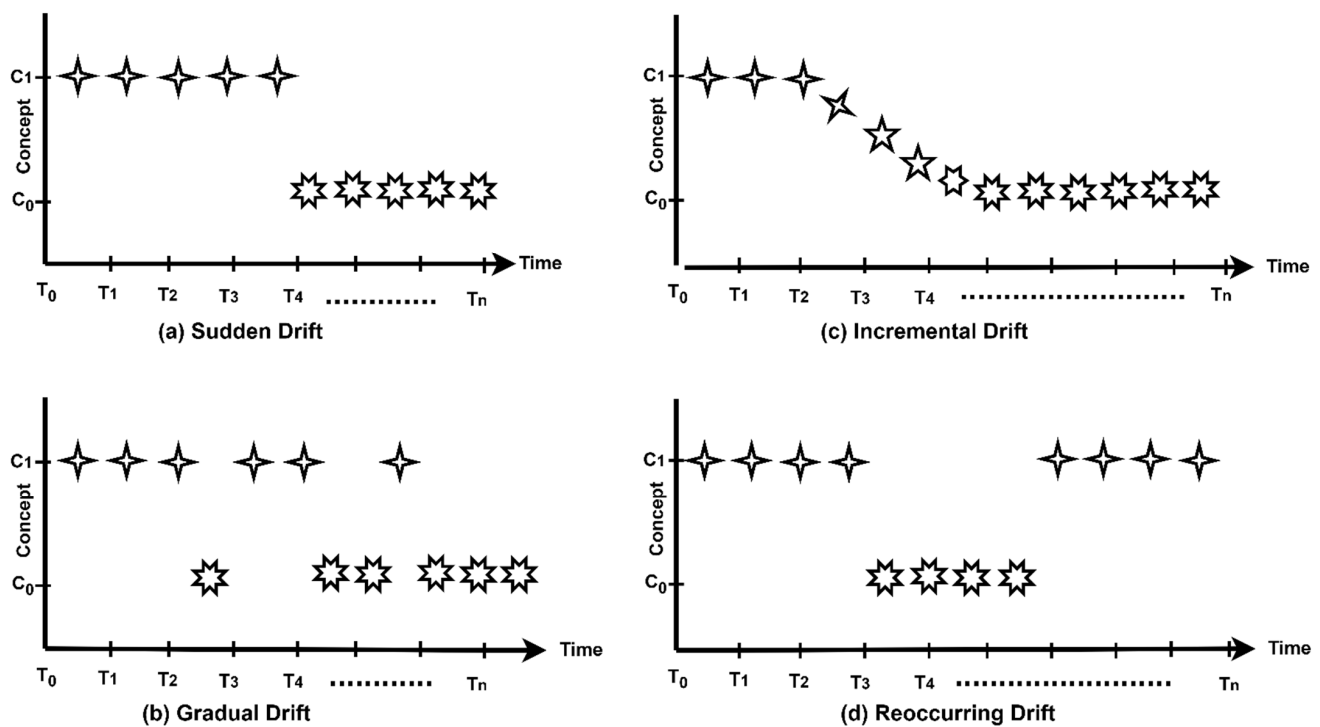


Figure 1. Types of concept drift based on time [17,18].

Besides the drift, the change may occur in class distribution, and the number of instances of each class may also vary, leading to a class imbalance problem [15,19–21]. In case the stream is multi-class, the changes may occur in multiple classes simultaneously; this becomes more challenging for the drift detection methods to detect the new concept and adapt the existing ML model to the latest concept [22]. When data face a class imbalance problem, it signifies that the data are not equally distributed among classes. Some of the classes may have a very high number of instances called majority classes, and on the contrary, some have very skewed data or a smaller number of instances called minority classes. An ML model which is trained on imbalanced data produces biased results toward the majority class and misclassifies the minority class [23–25]. Along with that, in non-stationary multi-class streams, the majority class may become the minority class and vice versa [26].

In situations where concept drift and class imbalance both appear concurrently, the drift detection method and data balancing approaches are affected as well as the ML classification model. There is a possibility that only the class imbalance ratio alters at some point in time, but the concept remains the same. In such a situation, the distribution-based drift detection method may trigger a ‘drift’ alarm which is a false alarm. Similarly, the error rate-based drift detection methods monitor the performance of the ML model and may also generate false alarms in case the performance of ML model is decreased due to class imbalance.

Therefore, this study experimentally analyzes the existing drift detection methods in presence of multi-class imbalanced data. One of the prime objectives of this study is to investigate and find the best existing approach for drift detection in multi-class data streams in the presence of class imbalance and different drift types such as sudden, gradual, and incremental drift.

The foremost contributions of the study are the following:

- Highlighting the effect of concept drift and class imbalance in a multi-class data stream environment.
- Simulation of multi-class data streams with different drift types and class imbalance.
- Analysis of existing 10 state-of-the-art drift detection methods.
- Highlighting the performance difference of an ML model on balanced and imbalanced data streams in presence of concept drift (sudden, gradual, and incremental).

The rest of the paper is organized as follows: Section 2 highlights the existing work on drift detection methods. Section 3 explains the generated data streams, drift detection methods evaluated, and the performance measures used in this study. Section 4 discusses the results, and Section 5 concludes the overall study.

2. Related Work

The major goal of concept drift detection methods is to provide an efficient approach that works together with the classification model, indicating drift or novelty when there is a substantial change in data properties [27]. The model is updated in this manner, avoiding that it be influenced by the change and so boosting its predicted performance.

The existing drift detection methods are generally divided into three categories which are error rate-based, data distribution-based, and multiple hypothesis test-based or statistical test-based [18]. The error rate-based drift detection methods continuously monitor the performance of the base classifier. The base classifiers are used to categorize incoming instances in concept drift detection systems. It generates a class prediction for each instance, which is then compared to the actual class label. The drift detection technique states that whether or not a drift has occurred based on the classification result. In the end, the instance is used to train the base classifier. Every time a new instance arrives, this procedure is repeated. The Drift Detection Method (DDM) [28] is a well-known example of an error rate-based approach. It increases or decreases the value or error rate based on the classification performance of the base classifier. In the case of correct classification of the incoming instance by the base classifier, DDM decreases the value of the error rate and increases otherwise. When the level of error rate reaches a certain threshold, DDM either considers it as a drift or generates a warning. The DDM produced good results for sudden and gradual drift but not for the slow, gradual drifts.

The Early Drift Detection Method (EDDM) [29], an improved version of DDM, was proposed to increase the performance of slow progressive drifts. Instead of only the error rate, the EDDM tracks the average distances of two errors as well as the two running averages of standard deviations. The EDDM, like the DDM, has two threshold values to designate the warning zone and the drift zone. The reactive Drift Detection Method [30] (RDDM) periodically reduces the number of instances of a very long concept and recalculates the DDM stats to overcome the performance loss issue of DDM. The ADWIN [31] algorithm uses sliding windows to compare the mean of the output sequences that may vary with time. By analyzing the average of a statistic across the two windows, an algo-

rithm will determine the window's size and shape. Whenever the absolute value of the difference between the two averages exceeds a certain threshold, a change is recognized, and it resets the output windows.

The data distribution-based drift detection methods mostly use sliding windows to keep the recent data samples and apply statistics to identify the dissimilarity between two distributions in old and new data samples. Most traditional concept drift detection approaches combine learner (classifier) the current distribution and the dissimilarity measurement method (drift detector) to determine modifications and changes between the current and previous distributions. Distribution-based approaches may identify both real and virtual drifts, depending on the kind of dissimilarity metrics used. They may, however, be unable to discriminate the kinds of observed drifts [32]. Dasu et al. [33] presented a distribution-based drift detection method called kdqTree. This approach divides the distributions into smaller substructures. The substructures from the prior and most current data windows are compared using the Kullback–Leibler distance matrix to assess the drift. The study [34] proposes a concept drift detection approach that is based on the premise that there is restricted access to class labels or no access at all. When applied to unlabeled data streams, the suggested technique identifies concept drift based on the class label information that has been anticipated by the base classifier. The Geometric Moving Average [35] (GMA) gives the most recent observation the greatest weight, and all previous observations' weights decrease in geometric progression. A novelty detection algorithm in multi-class streams MINAS was proposed in [27], which separates the learning process into two stages: offline and online. The offline phase trains a decision model based on the problem's known concept. It is only used once. Following that, the online phase collects new samples and categorizes them as belonging to one of the existing classes or as unknown. In addition, the algorithm seeks a coherent collection of unknown samples to find new classes or expansions of established classes. These additional classes and extensions are included in the decision model. Furthermore, the approach enables the model to ignore out-of-date data and automatically adjust to concept drift.

Page Hinckley (PH) [36] is a sequential analysis test that monitors the difference between current accuracy and means accuracy of the base classifier for a current moment. If the difference between two accuracies reaches a threshold, it is considered as a drift. The Page–Hinckley test was used in [37] to detect the change in normal and abnormal signals. CUSUM [38] is a variant of PH which is a sequential analysis that uses the principles of the Sequential Probability Ratio Test (SPRT). It monitors the mean of the predicted result values and generates the drift alarm if the mean values reach to a certain threshold.

The hypothetical test-based drift detection can be applied to input data as well as the output of the ML model. The HDDM_A and HDDM_W were proposed in [39], which are similar to the DDM but use Hoeffding's inequalities. HDDM_A calculates the moving average, and HDDM_W calculates the weighted moving average from the incoming input stream. They first identify the relevant cut point in the incoming stream based on that they either apply A-test or W-test to find the current state of the stream. The possible options for stream state are stable, warning, or drift. A Statistical Test of Equal Proportions [40] (STEPD) computes the accuracy of the base classifier in the W most recent instances and compares it to its overall accuracy from the beginning of the learning process.

Recently, a window-based drift detection approach was proposed [41]. It has ability to store the most recent samples on a sliding window and removes the old ones. The samples of the sliding window are distributed in two sub-windows. One sub-window keeps the most recent samples from the main sliding window and considers it a new concept, whereas the other sub-window keeps the older samples and considers them as an old concept. They further apply the Kolmogorov–Smirnov [42] test to compare the absolute distance of the two sub-windows over a cumulative distribution. A window is a buffer memory to store limited data. This window can be used to store input data or output of the ML model to monitor the behavior of the data or output. Another sliding window-based drift detection method called One-Class Drift Detector (OCDD) was proposed in [43]. At

the start, a one-class classifier is trained using the data in the sliding window. The one-class classifier is used to estimate the distribution of the new concept, identifying whether fresh samples are from the existing idea or are outliers. Outlier samples are recognized as data from the new concept. It indicates a drift based on the proportion of outliers discovered in the sliding window. This technique is repeated indefinitely until there is no fresh data. OCDD's fundamental flaw is that it identifies drifts redundantly (false positives) when there are virtual drifts in which $P(X)$ changes, but $P(y|X)$ does not. The classifier is excessively adjusted, and the model loses important knowledge. Furthermore, it is incapable of detecting concept drifts (false negatives) in which changes occur solely on $P(y|X)$. In general, every drift detection method works either on the input data (monitors dissimilarity in data distribution) or on the output of the ML model (drop in accuracy or increase in error rate). We diagrammatically illustrated the input-based drift detector and output-based drift detector in Figure 2a,b.

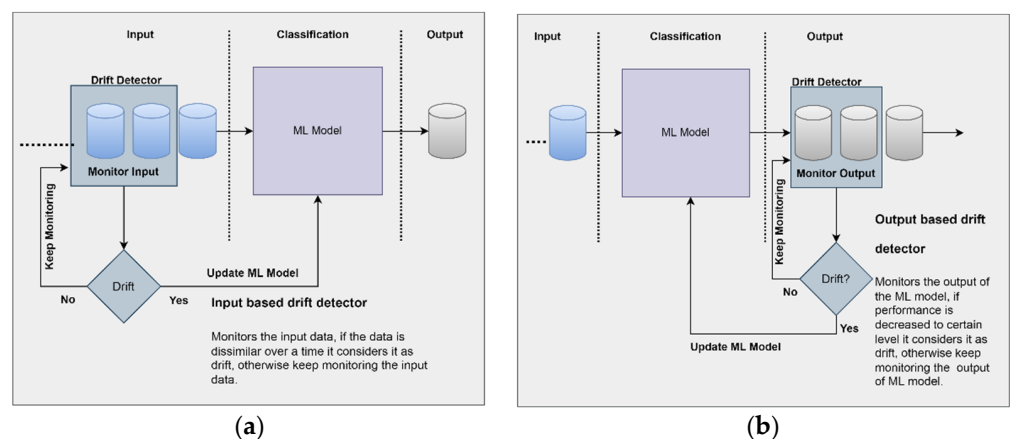


Figure 2. (a) Input-based drift detection; (b) output-based drift detection.

Most of the work presents a variety of drift detection methods in general. Therefore, the experimental study [44] examined eight drift detection approaches in terms of how they perform in the context of sudden and gradual concept drifts. Another work presented in [45] compared four drift detection methods on drifted data streams with noise. The study [46] compared fourteen drift detection methods on binary class data streams. Again, the sudden and gradual drift types were considered in these studies. These experimental studies are limited to sudden and gradual drift types only, and the data streams used in their study were mostly binary classes. Another main limitation is that the effect of the class imbalance issue was not considered in their work. Hence, comprehensive work is required which compares the existing drift detection methods on data streams with various drift types, in presence of a multi-class imbalance problem. A summary of the existing experimental studies which compare drift detection methods is given in Table 1.

Table 1. Comparative studies on drift detection methods.

Study	Concept Drift				Multi-Class Classification	Class Imbalance
	Sudden	Gradual	Incremental	Reoccurring		
[44]	✓	✓	×	×	×	×
[45]	✓	✓	×	×	✓	×
[46]	✓	✓	×	×	×	×
Ours	✓	✓	✓	To some extent	✓	✓

3. Methodology

Concept drift and class imbalance are common problems of applications that generate continuous non-stationary data streams. In real-time data streams, it is hard to know where exactly the drift has occurred, and as a result, drift detection methods cannot be properly tested in such situations. Therefore, we generate synthetic data streams to simulate the features of real-time data streams with specific and controllable drift types as carried out by [18,47]. The following subsections discuss the data generation mechanism and experimental setup in detail.

3.1. Data Stream Simulation

The study at hand focuses on testing the existing drift detection methods on multi-class balanced and imbalanced data streams with sudden, gradual, and incremental drifts. Therefore, to know where exactly the drift is occurring and what is the actual number of samples in each class, i.e., the class imbalance ratio, we simulated different data streams in a way that each stream had two versions, the balanced data stream and the imbalanced data stream. The reason for creating two versions of the same stream is to identify the performance difference between drift detection methods on balanced and imbalanced data streams when the drift type is the same. It is because we are uninformed of the drift location and drift type in real data streams that evaluating the performance of drift detection methods on such data streams is difficult [48]. As a result, for our investigations, synthetically produced data streams are employed in which we know the drift locations, drift type, and class imbalance ratio at the time of drift occurrence. We used a massive online analysis (MOA) [49] framework to generate the streams. The RandomRBFGenerator, RandomTreeGenerator, and LEDGenerator classes were used to generate different multi-class data streams. Every stream contains 100K samples, after every 25K samples, a new concept replaces the old one. The ‘sudden’ and ‘gradual’ drifts are easy to generate using the width (w) property. For sudden, we set $w=1$, and for gradual $w = 5K$, which means during these 5K instances, the concept will change gradually. The position (p) is set as 25K in all cases, which means it is the center of each drift (i.e., concept drift occurs after every 25K). Table 2 shows the summary of the generated streams.

Table 2. Summary of generated data streams.

Stream	Generator	Drift	CIR	C	A	I
Balance_SSS	RRBF, RT	S,S,S	2/2/2/2/2, 2/2/2/2/2, 2/2/2/2/2	5	21	100K
Imbalance_SSS	RRBF, RT	S,S,S	5/4/3/2/1, 5/4/3/2/1, 5/4/3/2/1	5	21	100K
Balance_GGG	RRBF, RT	G,G,G	2/2/2/2/2, 2/2/2/2/2, 2/2/2/2/2	5	21	100K
Imbalance_GGG	RRBF, RT	G,G,G	5/4/3/2/1, 5/4/3/2/1, 5/4/3/2/1	5	21	100K
Balance_III	RRBF, RT	I,I,I	2/2/2/2/2, 2/2/2/2/2, 2/2/2/2/2	5	21	100K
Imbalance_III	RRBF, RT	I,I,I	5/4/3/2/1, 5/4/3/2/1, 5/4/3/2/1	5	21	100K

CIR = Class Imbalance Ratio; C = Classes; A = Attributes; I = Instances.

3.2. Experimental Setup

The purpose of this study is to compare the existing drift detection methods on balanced and imbalanced multi-class data streams and find which of the compared methods contributes more to improve the performance of the base classifier and in what circumstances. Ten well-known drift detection methods are scientifically compared in this study on data streams with various drift types. In these experiments, we used the No Change Detection (NoCD) option to distinguish the performance of the base classifier without using any drift detector. It is to analyze the performance of the base classifier with a drift detector and without a drift detector. All these drift detection methods are used with default parameters, and no parameter tuning is performed. In real-time data stream classification data arrives so fast and may change the concept frequently. Therefore, if a model's parameters are tuned for the initial concept, they lose their worth if the concept changes. The summary of these methods is given in Table 3. All the experiments are performed in MOA [49], using a machine-equipped 2.3 GHz Intel i5 processor and 8 GB RAM. The results are evaluated in terms of the F-1 score metric which provides a way to combine both precisions and recall into a single measure that captures both properties. For data streams where the drift location is known such as streams with sudden drift, the methods are also evaluated using correct identification, distance from the actual drift location and identified location, false detection, and miss detection. The Hoeffding Tree classifier [50] is used as the base classifier because it is an incremental learner capable of learning from massive data streams. Hoeffding Tree with its incremental decision tree approach is capable of updating existing trees only utilizing the single input instance, eliminating the requirement to evaluate older instances.

Table 3. Summary of drift detection methods considered for analysis.

Sr	Method	Full Name	Technique
1	DDM [28]	Drift Detection Method	Increases or decreases the value or error rate based on the classification performance of the base classifier
2	EDDM [29]	Early Drift Detection Method	Averages distances of two errors, running averages of standard deviations
3	PH [36]	Page Hinckley	Monitors the difference between current accuracy and mean accuracy
4	RDDM [30]	Reactive Drift Detection Method	Recalculates the DDM stats in the current concept is too long (exceeds the defined maximum value)
5	CuSUM [38]	Cumulative Sum Drift Method	Monitors the mean of data. Considers a change when the mean reaches a certain threshold
6	GMA [35]	Geometric Moving Average	Gives the most recent observation the greatest weight, and all previous observations' weights decrease in geometric progression
7	EWMA [51]	Exponentially weighted moving average	Monitors the moving average of the observations, and decreases the weights of the old observations exponentially
8	HDDM_A [39]	Hoeffding's Drift Detection Method-Average	Monitors the performance metrics measured during the learning process using Hoeffding's bounds (moving average)
9	HDDM_W [39]	Hoeffding's Drift Detection Method-Weighted average	Monitors the performance metrics measured during the learning process using Hoeffding's bounds (weighted moving average)
10	STEPD [40]	Statistical Test of Equal Proportions	Computes the accuracy of the base classifier in the W most recent instances and compares it to its overall accuracy from the beginning of the learning process.

4. Results and Discussion

The main purpose of drift detection methods is to timely identify the drift in and trigger the ML model which is employed as the base classifier to adjust itself accordingly so that the classification performance degradation can be avoided. Keeping this in view, the predictive performance of the classifier has assessed when combined with each of the 10 drift detection methods. It is hypothesized in the current study that the performance of the base classifier will be improved using drift detection methods. The experiments were performed on data streams with sudden drift, gradual drift, and incremental drift. We discuss the results for each drift type separately in the following sub-sections.

4.1. Balanced and Imbalanced Data Stream with Sudden Drift

Results of the experiments in terms of F1-score on balanced and imbalanced data streams are presented in Figure 3. It can be observed from the results that the base classifier has produced better results on balanced data streams compared to the imbalanced data stream. Another notable aspect to observe is that the performance of the base classifier without the drift detection method (NoCD) produced far better results than using drift detection methods except GMA. In other words, instead of performance improvement, the performance of the base classifier has been decreased using most of the compared drift detection methods.

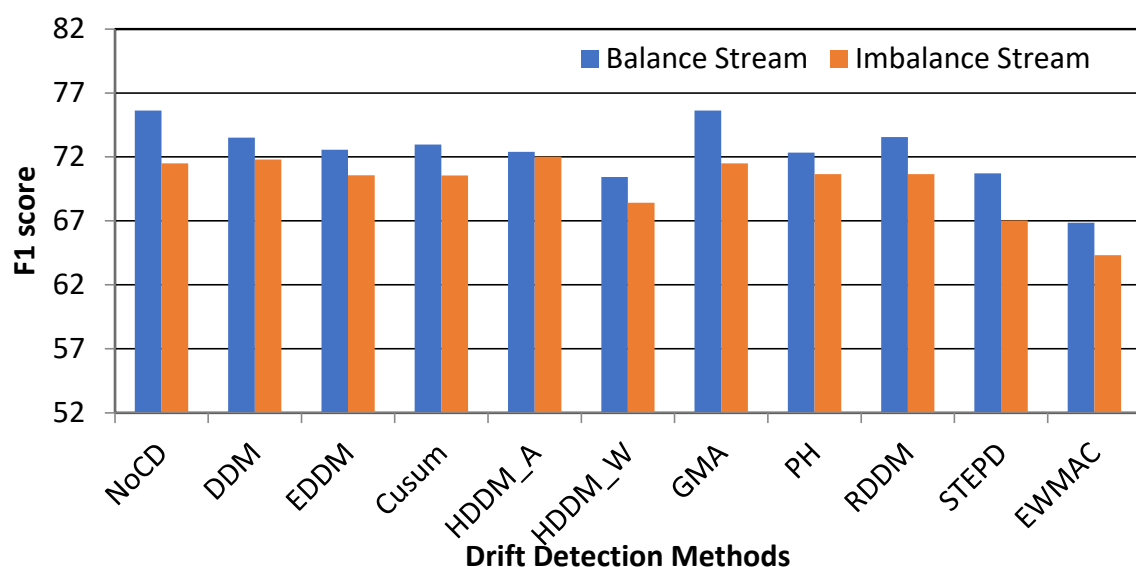


Figure 3. F1 score on balanced and imbalanced data streams with sudden drift.

Figure 4 shows the actual effect of each drift detection method on base classifier performance in terms of percentage increase or decrease. On the y-axis if the value is <0 , it means the performance is decreased, and >0 shows the performance improvement, whereas 0 shows no effect. The performance of HT as a base classifier on balanced and imbalanced data streams was decreased using most of the drift detection methods. The performance of the base classifier with GMA remained unchanged, i.e., neither increased nor decreased on balanced as well as imbalanced data streams. Whereas, with DDM and HDDM_A, the performance was slightly improved on the imbalanced data stream.

These results raise the question of why the performance of the base classifier is decreasing using drift detection methods instead of improving. It has been analyzed from another perspective, i.e., the number of false drift alarms. A false alarm takes place when the detector signals a change when there were none. So, when the drift detection method generates the false alarm, it triggers mechanism to reset the base classifier. As a result, the base classifier forgets the previously learned knowledge. In case there is no drift in the data streams and the concept is still the same and the drift detection method generates

a false alarm (considering that there is a drift), the base classifier will be reset and starts learning from the new concept but in real concept is still the same. This situation causes performance degradation. Therefore, any drift detection method which generates a high number of false alarms causes a decrease in performance. Figure 5 shows the number of false alarms generated by each drift detection method on the balanced and imbalanced data streams. It can be observed that HDDM_W, STEPDP, and EWMAC generate a high number of false alarms, and these are the methods that cause more damage to the base classifier's performance. On an imbalanced data stream, both DDM and HDDM_A methods correctly detected actual drift. They did not generate many false alarms, and as a result, they slightly improved the performance of the base classifier.

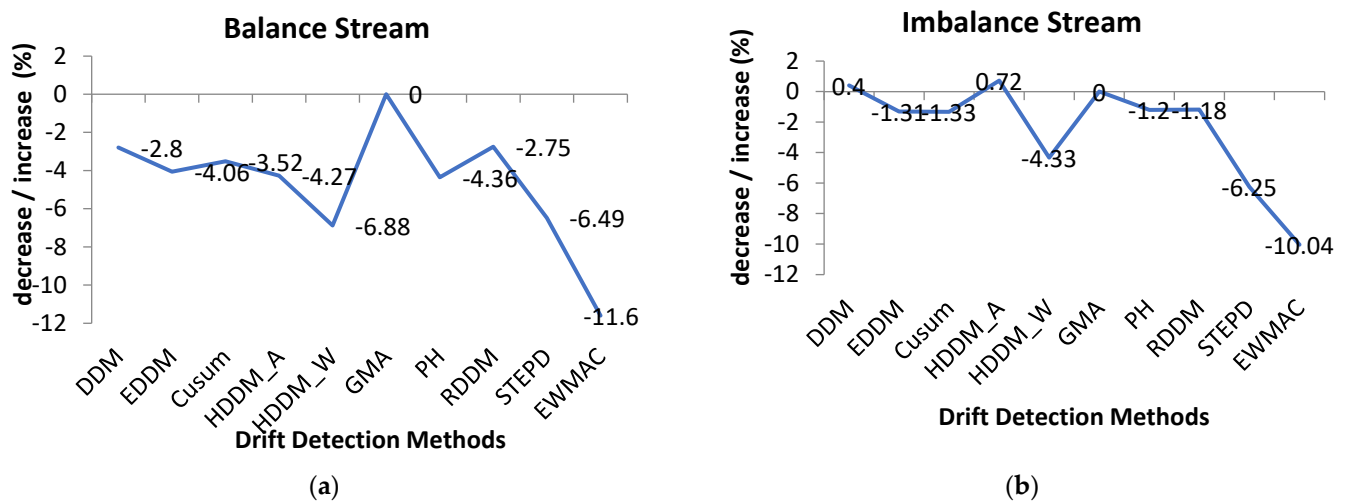


Figure 4. Effect of drift detection methods on base classifier's performance: (a) sudden drift without class imbalance; (b) sudden drift with class imbalance.

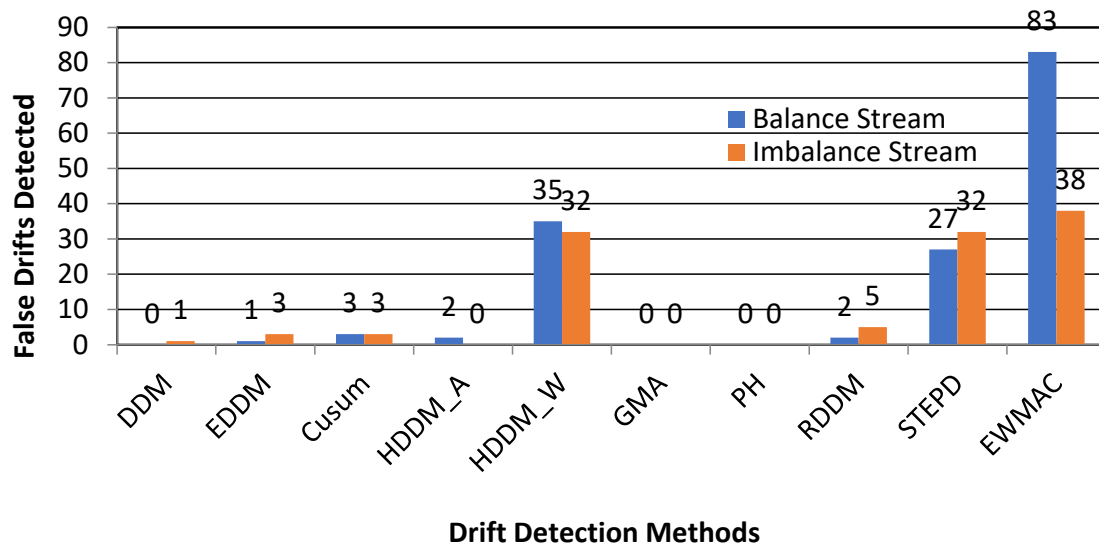


Figure 5. Number of false drifts detected by each method on balanced and imbalanced data streams.

The performance of drift detection methods was also measured on a few other metrics such as True Detection, False Alarm, and Delay in Detection, which are described in [23].

The results are presented in Table 4. As can be seen in Table 4, the value 0 in column ‘Drift Detected’ against GMA highlights that it did not identify any of the drift out of three drifts available in the data stream; hence, it missed all 3 drifts. The distance (T) represents the distance in time stamp from the actual drift location to the drift detection location, whereas the distance (I) is the distance (in a number of instances) from the first instance of the new concept to the instance where drift is detected by the drift detection method. Any drift detection method which produces poor results on any of the metrics in Table 4 will affect the performance of the base classifier.

From Table 4, it can be concluded that false drift detection is the main factor that causes the decrease in the performance of a base classifier. The EWMAC correctly detected all three drifts without any delay, but it also detected a high number of false drifts and detected 83 false drifts. Because of the false detection, the classifier is reset every time, and as a result, it forgets previously learned knowledge, which caused the 11.6% decrease in base classifier performance. HDDM_W and STEPD also correctly detected all three actual drifts but detected 35 and 27 false drifts, respectively. As a result, HT performance was decreased by 6.88% using HDDM_W and 6.49 using STEPD as a drift detection method.

The EDDM causes a 4.07% performance decrease in HT. However, it detected one false drift, missed 2 actual drifts, and detected just one actual drift but with a delay. These factors contributed to the performance degradation of EDDM.

As observed, false drift detection is the major factor that causes the performance degradation of the base classifier. The reason behind the false drift detection rate is the technique applied by the underlying method to detect the drift. For example, the EWMAC-DM exponentially decreases the weight of the old data. Hence, after a certain period, the EWMAC-DM may consider the new data belonging to a new concept even if that belongs to the current concept.

Table 4. Performance comparison on other metrics.

Method	True Detection		Drifts Missed		Distance (T)		Distance (I)		False Alarm	
	Bal.	Imb.	Bal.	Imb.	Bal.	Imb.	Bal.	Imb.	Bal.	Imb.
DDM	3	3	0	0	0.33	0	165	0	0	1
EDDM	1	1	2	2	1.67	1.333	835	666.65	1	3
Cusum	3	3	0	0	0	0	0	0	3	3
HDDM_A	3	3	0	0	0	0	0	0	2	0
HDDM_W	3	3	0	0	0	0	0	0	35	32
GMA	0	0	3	3	0	0	0	0	0	0
PH	3	3	0	0	2.33	2	1165	1000	0	0
RDDM	3	3	0	0	0	0	0	0	2	5
STEPD	3	3	0	0	0	0	0	0	27	32
EWMAC	3	2	0	1	0	2	0	1000	83	38

Bal. = Balanced (data stream); Imb. = Imbalanced (data stream).

4.2. Balanced and Imbalanced Data Stream with Gradual Drift

The gradual drift is the change of one concept to another gradually. In this study, this change is completed in 10 timestamps (5000 instances). The performance of the base classifier on balanced and imbalanced multi-class data streams is presented in Figure 6. Compared to the data stream with sudden drift, where the base classifier was able to produce good results without using the drift detection methods, and its performance was decreased with most of the drift detection methods. In the case of the data stream with gradual drift, some of the drift detection methods have contributed to improving the performance of the base classifier.

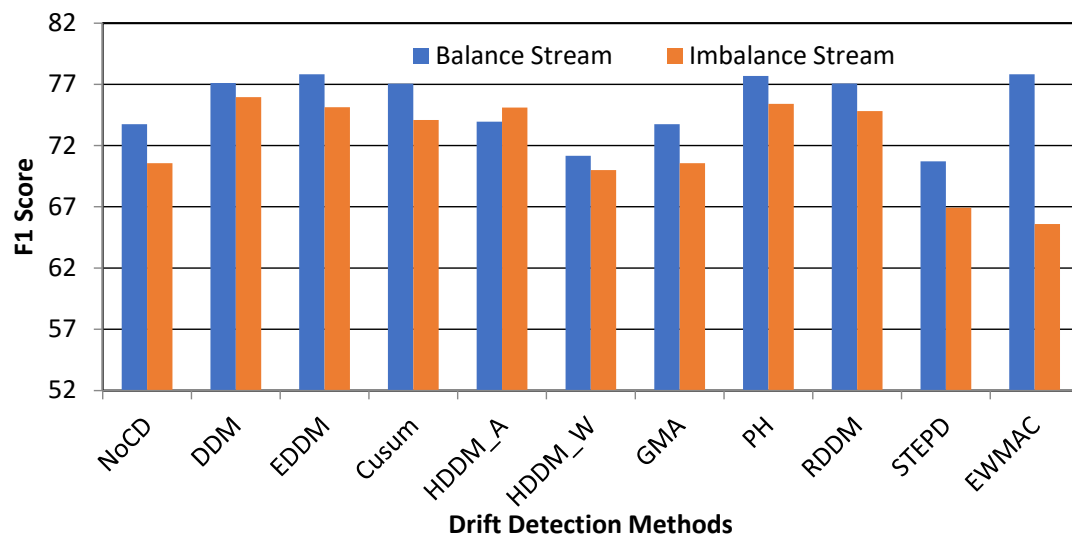


Figure 6. F1 score on balanced and imbalanced data streams with gradual drift.

Figure 7 shows the actual impact on the classifier's performance in percentage. On a balanced data stream, the performance of the base classifier is improved using DDM, EDDM, CuSUM, PH, RDDM, and EWMAC by 4.35%, 5.22%, 4.28%, 5.06%, 4.3%, 5.22%, and 6.22%, respectively, whereas HDDM_W and STEPDP cause a decrease in the performance of the base classifier by 3.63% and 4.31%. The GMA again showed no effect on the base classifier's performance on both streams. On an imbalanced data stream with gradual drift, using DDM, EDDM, CUSUM, PH, and RDDM contributed to improving the base classifier's performance, whereas EWMAC badly affected the classifier's performance when the data stream had a class imbalance issue.

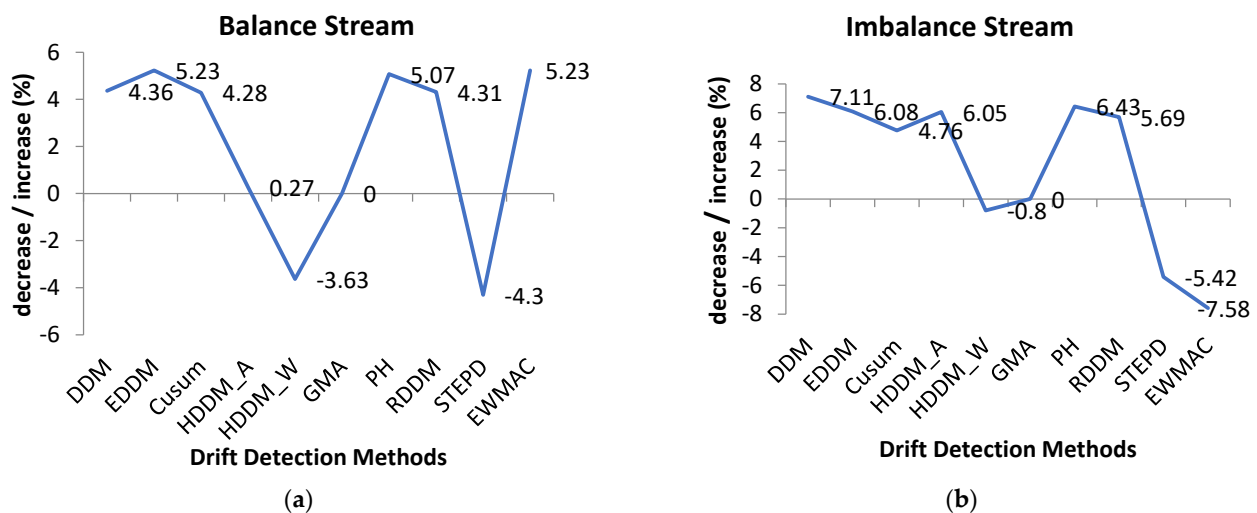


Figure 7. Effect of drift detection methods on base classifier's performance in case of gradual drift. (a) Gradual drift without class imbalance; (b) Gradual drift with class imbalance.

Again, we looked into the false alarm factor, as we did for streams with sudden drift. Figure 8 shows the number of false alarms generated by each drift detection method on the balanced and imbalanced data stream. It can be observed that again HDDM_W, STEPDP, and EWMAC generated a high number of false alarms for both streams so as they affected the base classifier's performance. Even though EWMAC generated 63 false alarms but still produced on the balanced data stream, on the imbalanced data stream, the false alarm rate was increased to 87 and which causes the decrease in the base classifier's

performance. From Figure 8, it can also be observed that DDM, EDDM, CUSUM, PH, and RDDM generated either 0 or very few false alarms; hence, they contributed to improving the base classifier's performance.

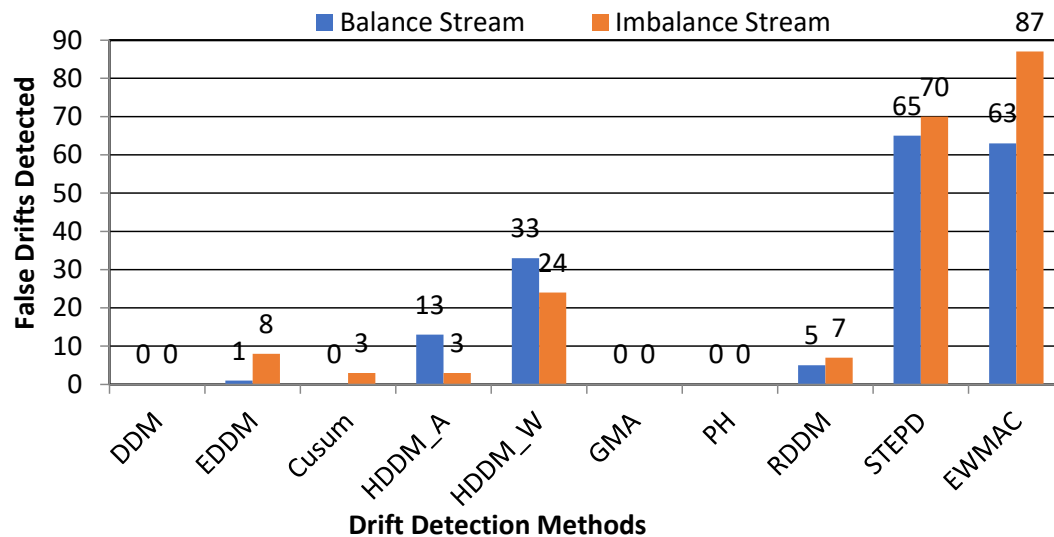


Figure 8. Number of false drifts detected by each method on balanced and imbalanced data stream.

4.3. Balanced and Imbalanced Data Stream with Incremental Drift

An incremental drift takes place when the concept changes gradually with a constant speed, and this is the most difficult drift to detect. The performance of the base classifier on balanced and imbalanced multi-class data streams with incremental drift is presented in Figure 9. It can be observed that it is hard to learn from the data stream with incremental drift and the classifier has produced poor results as compared to streams with gradual drift and streams with sudden drift. The class imbalance further decreases the performance compared to the balanced data stream.

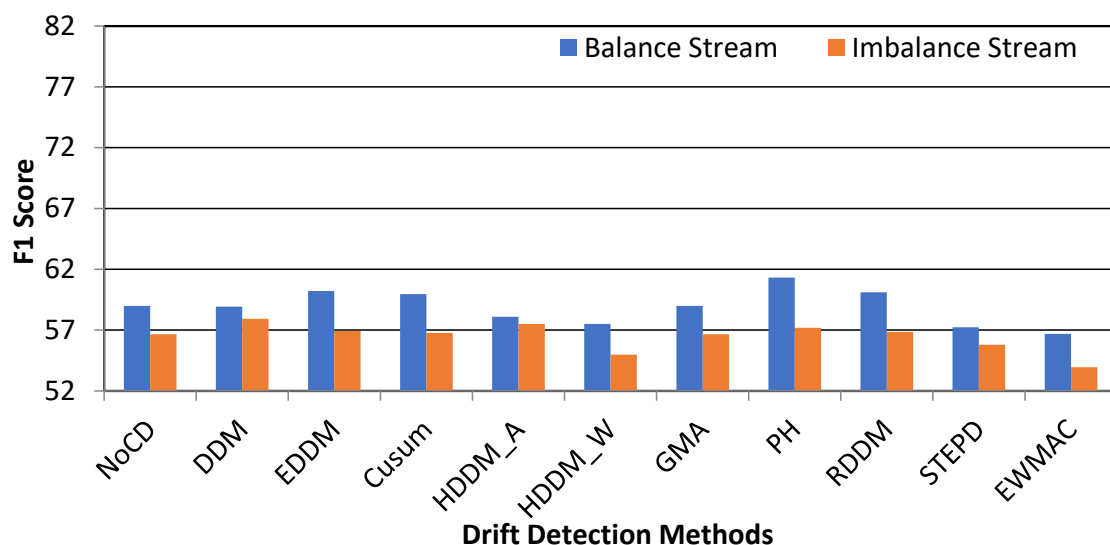


Figure 9. F1 score on balanced and imbalanced data streams with incremental drift.

Figure 10 shows the actual effect of each drift detection method on base classifier performance in terms of percentage increase or decrease. The performance of HT as a base classifier on balanced and imbalanced data streams was again decreased using HDDM_W, STEP, and EWMAC. The HDDM_W also caused a decrease in performance

on the balanced data stream. The performance of the base classifier with GMA remained unchanged again, i.e., neither increased nor decreased on balanced as well as imbalanced data streams. Whereas, with DDM and HDDM_A, the performance was slightly improved on an imbalanced data stream. The DDM and HDDM_A cause a decrease in performance whereas they produced an improved performance on the imbalanced data stream. Overall, PH produced a high F1 score on a balanced data stream with incremental drift whereas DDM produced a high F1 score on an imbalanced data stream with incremental drift. For streams with incremental drift, we cannot calculate the false alarms because the position of increment drift is always unknown.

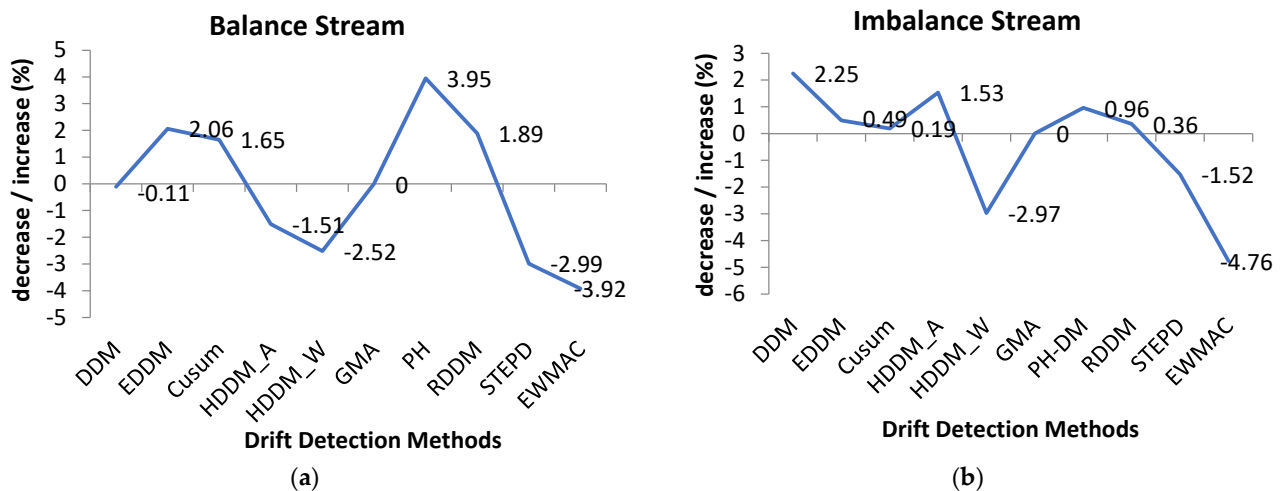


Figure 10. Effect of drift detection methods on base classifier's performance in case of incremental drift: (a) incremental drift without class imbalance; (b) incremental Drift with class imbalance.

The overall performance of the base classifier on each data stream is presented in Figure 11a–f. It can be observed that the performance of the base classifier is decreased at the time of drift occurrence. Another point to observe here is that class imbalance has further affected the performance of most drift detection methods. Table 5 shows the performance in terms of the overall F1 score on each data stream. The bold values represent the best performances on a particular data stream. The results show that none of these methods has shown consistent performance on all data stream types.

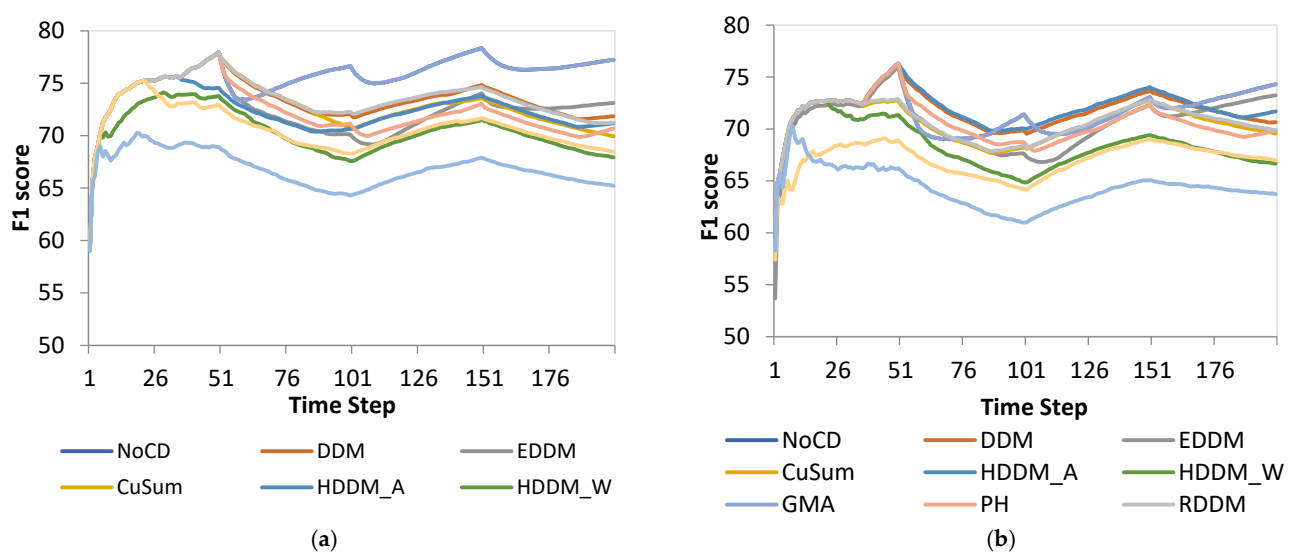


Figure 11. Cont.

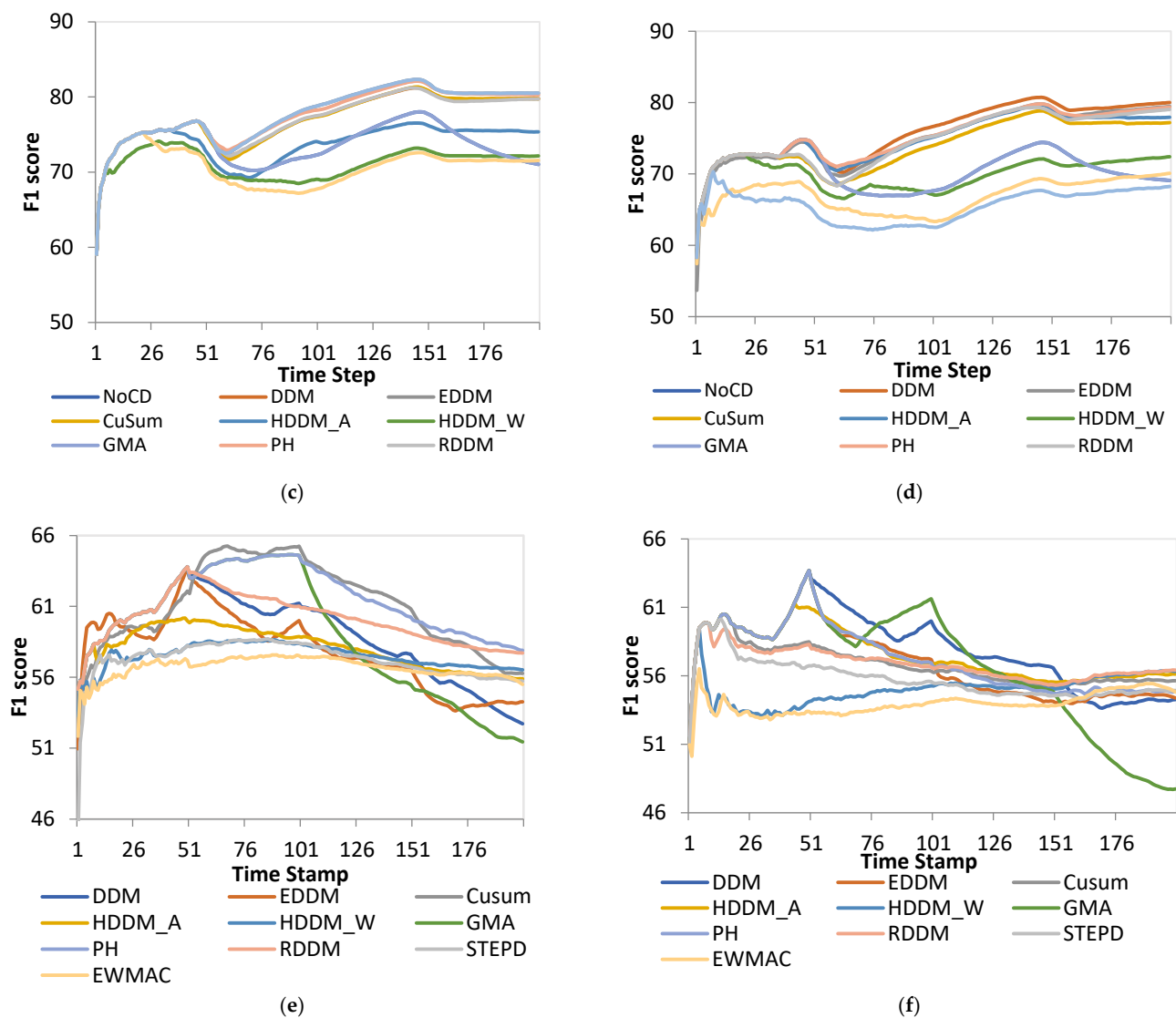


Figure 11. F1 score on each time step for all data streams: (a) performance of HT on balanced data with sudden drift; (b) performance of HT on imbalanced data with sudden drift; (c) performance of HT on balanced data with gradual drift; (d) performance of HT on imbalanced data with gradual drift; (e) performance of HT on balanced data with incremental drift; (f) performance of HT on imbalanced data with incremental drift.

Table 5. Overall F1 score of HT classifier with different drift detection methods.

DS	NoCD	DDM	EDDM	Cusum	HDDM_A	HDDM_W	GMA	PH	RDDM	STEPD	EWMAC
B-S	75.6318	73.5086	72.5561	72.9715	72.4036	70.4254	75.6318	72.3253	73.5527	70.7199	66.8582
I-S	71.501	71.7858	70.5645	70.5466	72.0156	68.4061	71.501	70.6462	70.6549	67.034	64.3249
B-G	73.7541	77.1057	77.8197	77.0541	73.9509	71.1709	73.7541	77.6869	77.0689	70.7089	77.8197
I-G	70.555	75.9551	75.1269	74.0852	75.0977	69.9959	70.555	75.4099	74.8177	66.9283	65.5907
B-I	58.9903	58.9247	60.2054	59.9613	58.0969	57.5035	58.9903	61.3183	60.1062	57.228	56.6787
I-I	56.6491	57.9251	56.9282	56.7571	57.5149	54.966	56.6491	57.1908	56.8511	55.7872	53.9512

DS: Data Stream; **NoCD:** No Change Detection Method; **B-S:** Balanced Stream with Sudden Drift; **I-S:** Balanced Stream with Sudden Drift; **B-G:** Balanced Stream with Gradual Drift; **I-G:** Imbalanced Stream with Gradual Drift; **B-I:** Balanced Stream with Incremental Drift; **I-I:** Imbalanced Stream with Incremental Drift.

Therefore, from Table 5, we generated the ranking of the methods on each data stream and calculated the overall ranking same can be seen in Table 6. In this study, one base classifier is tested with 10 drift detections and one without the drift detection method, so the best ranking is 1, and the worst ranking is 11. The rankings were calculated using the RANK.EQ function available in Microsoft excel. The results show that the DDM produced the best average ranking which is 3.17. Both EDDM and PH produced the 2nd best average ranking with 3.83. RDDM produced the 3rd best average ranking with 4.33. 4, respectively. The STEP_D, EWMAC, and HDDM_W remained the poorest performers with 9.83, 9.33, and 9.5, respectively. On imbalance data streams specifically, the DDM produced the best ranking, i.e., ranked 2nd on I-S, ranked 1st on I-G, and ranked 1st on I-I. The ranking of GMA is the ranking of the base classifier because GMA showed no effect on the performance of HT as the base classifier.

Statistical analysis: Results of all drift detection methods are statistically analyzed to identify the performance significance. We used the Friedman test [52] to compare the results of all the methods investigated across six data streams. The Friedman test function `FriedmanchisquareResult` of python was used to run the test. Friedman's test result with statistic = 32.94 and $p = 0.00028$ rejects the null hypothesis (all drift detection methods produce the same performance) as $p < 0.05$. Still, it is not clear which of the compared methods performed significantly differently. To achieve this the Bonferroni–Dunn post hoc test [53] is used to determine which algorithms are statistically distinct. The result of the Bonferroni test is shown in Figure 12. The methods which are not significantly different (at $p = 0.05$) are connected.

Table 6. Overall ranking of HT classifier with different drift detection methods.

Datasets	NoCD	DDM	EDDM	CUSUM	HDDM_A	HDDM_W	GMA	PH	RDDM	STEPD	EWMAC
B-S	1	4	6	5	7	10	1	8	3	9	11
I-S	3	2	7	8	1	9	3	6	5	10	11
B-G	8	4	1	6	7	10	8	3	5	11	1
I-G	7	1	3	6	4	9	7	2	5	10	11
B-I	5	7	2	4	8	9	5	1	3	10	11
I-I	7	1	4	6	2	10	7	3	5	9	11
Sum of ranks	31	19	23	35	29	57	31	23	26	59	56
Average ranks	5.17	3.17	3.83	5.83	4.83	9.5	5.17	3.83	4.33	9.83	9.33

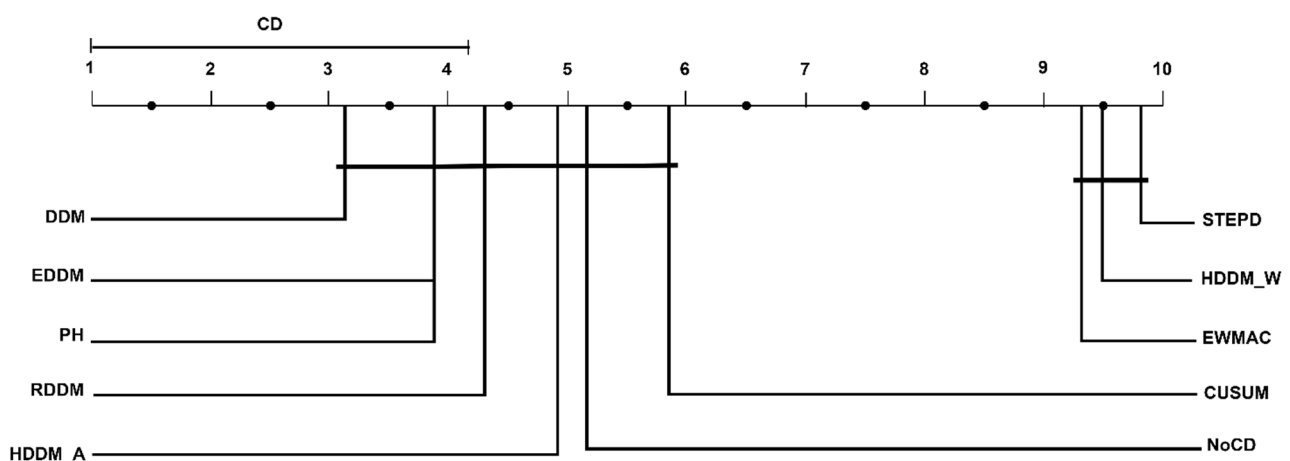


Figure 12. Comparison of drift detection methods against each other with the Bonferroni–Dunn test.

5. Conclusions

In this world of advancements, most industrial sensor-based applications generate continuous non-stationary data streams which have inbuilt issues of concept drift and class imbalance. Due to the change in concept, the performance of the output predicting model decreases. To counter this issue, a variety of drift detection methods are proposed to detect the concept drift and adapt the model to a new concept. These methods struggle when the data have a multi-class imbalance issue. To examine how these drift detection methods perform on multi-class data streams, this work compares 10 state-of-the-art drift detection methods on synthetic multi-class balanced and imbalanced data streams. We generated these data streams with sudden, gradual, and incremental drifts. The data stream also faces the issue of class imbalance, so two copies of each data stream were generated with the same type of concept drift. One with class imbalance (imbalanced data stream) and the other without class imbalance (balanced stream). The performance was evaluated in terms of the F1 score. The other performance measures were also used for sudden and gradual drifts such as false alarms and delays in drift detection.

This experimental study accomplishes the following results:

- Every drift type (sudden, gradual, or incremental) has a different effect on the performance of the base classifier.
- None of the compared methods showed consistent performance on different data streams.
- Most of the drift detection methods were affected by the class imbalance issue.
- Our results showed that DDM produced the best overall ranking among the compared methods but significantly different from STEPD, HDMM_W, and EWMAC only.
- DDM also produced great results on data streams with class imbalance.
- The STEP_D, EWMAC, and HDDM_W produced poor results on both balanced and imbalanced data streams.
- Our results also showed that the drift detection method may cause a decrease in base classifier performance in case it generates a false alarm.

Based on the results of this study, we can conclude that the existing drift detection methods are not only lacking in the detection of multiple drifts, but they also struggle when data face multi-class imbalance issues. Therefore, there is a need to design adaptive drift detection methods that can quickly learn the new concepts and update themselves accordingly without generating false alarms. This is a goal for our future work. We are also planning to test more existing drift detection methods as well as the latest methods on numerous multi-class data streams with dynamic class imbalance ratio.

Author Contributions: Conceptualization, A.S.P. and J.J.; methodology, H.M.G.; software, A.S.P.; validation, H.M.G., A.R.G. and M.A.H.; formal analysis, J.J.; investigation, A.S.P.; resources, A.S.P.; data curation, H.M.G.; writing—original draft preparation, A.S.P.; writing—review and editing, J.J. and A.R.G.; visualization, A.S.P.; supervision, J.J.; project administration, J.J.; funding acquisition, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by UTP URIF (grant number: 015LB0-053).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the Center for Research in Data Science and High-Performance Cloud Computing Centre (HPC3), Universiti Teknologi PETRONAS for their support to complete this research study.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wang, H.; Ma, X.; Zhao, Y. An improved Wiener process model with adaptive drift and diffusion for online remaining useful life prediction. *Mech. Syst. Signal Process.* **2019**, *127*, 370–387. [\[CrossRef\]](#)
- Wang, D.; Tsui, K.-L. Brownian motion with adaptive drift for remaining useful life prediction: Revisited. *Mech. Syst. Signal Process.* **2018**, *99*, 691–701. [\[CrossRef\]](#)
- Yu, W.; Tu, W.; Kim, I.Y.; Mechefske, C. A nonlinear-drift-driven Wiener process model for remaining useful life estimation considering three sources of variability. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107631. [\[CrossRef\]](#)
- Thiyagarajan, K.; Kodagoda, S.; Van Nguyen, L.; Ranasinghe, R. Sensor failure detection and faulty data accommodation approach for instrumented wastewater infrastructures. *IEEE Access* **2018**, *6*, 56562–56574. [\[CrossRef\]](#)
- Wu, Z.; Lin, W.; Ji, Y. An integrated ensemble learning model for imbalanced fault diagnostics and prognostics. *IEEE Access* **2018**, *6*, 8394–8402. [\[CrossRef\]](#)
- Zhong, K.; Han, M.; Han, B. Data-Driven Based Fault Prognosis for Industrial Systems A Concise Overview. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 330–345. [\[CrossRef\]](#)
- Zhou, F.; Yang, S.; Fujita, H.; Chen, D.; Wen, C. Deep learning fault diagnosis method based on global optimization GAN for unbalanced data. *Knowl.-Based Syst.* **2020**, *187*, 104837. [\[CrossRef\]](#)
- Qian, W.; Li, S. A novel class imbalance-robust network for bearing fault diagnosis utilizing raw vibration signals. *Meas. J. Int. Meas. Confed.* **2020**, *156*, 107567. [\[CrossRef\]](#)
- Wei, J.; Huang, H.; Yao, L.; Hu, Y.; Fan, Q.; Huang, D. New imbalanced fault diagnosis framework based on Cluster-MWMOTE and MFO-optimized LS-SVM using limited and complex bearing data. *Eng. Appl. Artif. Intell.* **2020**, *96*, 103966. [\[CrossRef\]](#)
- Sousa, M.R.; Gama, J.; Brandão, E. A new dynamic modeling framework for credit risk assessment. *Expert. Sys. Appl.* **2016**, *45*, 341–351. [\[CrossRef\]](#)
- Hegde, J.; Rokseth, B. Applications of machine learning methods for engineering risk assessment—A review. *Saf. Sci.* **2020**, *122*, 104492. [\[CrossRef\]](#)
- Chen, J.; Wu, Z.; Zhang, J. Driving Safety Risk Prediction Using Cost-Sensitive with Nonnegativity-Constrained Autoencoders Based on Imbalanced Naturalistic Driving Data. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4450–4465. [\[CrossRef\]](#)
- Talaei-Khoei, A.; Tavana, M.; Wilson, J.M. A predictive analytics framework for identifying patients at risk of developing multiple medical complications caused by chronic diseases. *Artif. Intell. Med.* **2019**, *101*, 101750. [\[CrossRef\]](#)
- Wolff, P.; Grana, M.; Ríos, S.A.; Yarza, M.B. Machine Learning Readmission Risk Modeling: A Pediatric Case Study. *BioMed Res. Int.* **2019**, *2019*, 8532892. [\[CrossRef\]](#)
- Khandekar, V.S.; Srinath, P. Non-stationary Data Stream Analysis: State-of-the-Art Challenges and Solutions. In Proceedings of the International Conference on Computational Science and Applications, Las Vegas, NV, USA, 16–18 December 2020; pp. 67–80.
- Ghazikhani, A.; Monsefi, R.; Yazdi, H.S. Ensemble of online neural networks for non-stationary and imbalanced data streams. *Neurocomputing* **2013**, *122*, 535–544. [\[CrossRef\]](#)
- Agrahari, S.; Singh, A.K. Concept Drift Detection in Data Stream Mining: A literature review. *J. King Saud Univ.—Comput. Inf. Sci.* **2021**, *in press*. [\[CrossRef\]](#)
- Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. [\[CrossRef\]](#)
- Ghazikhani, A.; Monsefi, R.; Yazdi, H.S. Online neural network model for non-stationary and imbalanced data stream classification. *Int. J. Mach. Learn. Cybern.* **2014**, *5*, 51–62. [\[CrossRef\]](#)
- Shen, L.; Er, M.J.; Liu, W.; Fan, Y.; Yin, Q. Population structure-learned classifier for high-dimension low-sample-size class-imbalanced problem. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104828. [\[CrossRef\]](#)
- Raghuwanshi, B.S.; Shukla, S. UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning. *Eng. Appl. Artif. Intell.* **2018**, *74*, 252–270. [\[CrossRef\]](#)
- Korycki, Ł.; Krawczyk, B. Concept Drift Detection from Multi-Class Imbalanced Data Streams. *arXiv* **2021**, arXiv:2104.10228.
- Wang, S.; Minku, L.L.; Yao, X. A systematic study of online class imbalance learning with concept drift. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4802–4821. [\[CrossRef\]](#)
- Kaur, H.; Pannu, H.S.; Malhi, A.K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.* **2019**, *52*, 1–36. [\[CrossRef\]](#)
- Haixiang, G.; Yijing, L.; Yanan, L.; Xiao, L.; Jinling, L. BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Eng. Appl. Artif. Intell.* **2016**, *49*, 176–193. [\[CrossRef\]](#)
- Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [\[CrossRef\]](#)
- Faria, E.R.; Gama, J.; Carvalho, A.C. Novelty detection algorithm for data streams multi-class problems. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 795–800.
- Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. Learning with drift detection. In Proceedings of the Brazilian Symposium on Artificial Intelligence, São Bernardo do Campo, Brazil, 23–28 October 2004; pp. 286–295.
- Baena-Garcia, M.; del Campo-Ávila, J.; Fidalgo, R.; Bifet, A.; Gavalda, R.; Morales-Bueno, R. Early drift detection method. In Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams, Pennsylvania, PA, USA, 6 August 2006; pp. 77–86.

30. Barros, R.S.M.; Cabral, D.R.L.; Gonçalves, P.M.; Santos, S.G.T.C. RDDM: Reactive drift detection method. *Expert. Sys. Appl.* **2017**, *90*, 344–355. [\[CrossRef\]](#)
31. Bifet, A.; Gavalda, R. Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007; pp. 443–448.
32. Tan, C.H.; Lee, V.C.S.; Salehi, M. Information resources estimation for accurate distribution-based concept drift detection. *Inf. Process. Manag.* **2022**, *59*, 102911. [\[CrossRef\]](#)
33. Dasu, T.; Krishnan, S.; Venkatasubramanian, S.; Yi, K. An information-theoretic approach to detecting changes in multi-dimensional data streams. In Proceedings of the Interface of Statistics, Computing Science, and Applications, Pasadena, CA, USA, 24–27 May 2006.
34. Kim, Y.; Park, C.H. An efficient concept drift detection method for streaming data under limited labeling. *IEICE Trans. Inf. Syst.* **2017**, *100*, 2537–2546. [\[CrossRef\]](#)
35. Roberts, S.J.T. Control chart tests based on geometric moving averages. *Technometrics* **2000**, *42*, 97–101. [\[CrossRef\]](#)
36. Page, E.S.J.B. Continuous inspection schemes. *Biometrika* **1954**, *41*, 100–115. [\[CrossRef\]](#)
37. Mouss, H.; Mouss, D.; Mouss, N.; Sefouhi, L. Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. In Proceedings of the 2004 5th Asian Control Conference (IEEE Cat. No. 04EX904), Melbourne, Australia, 20–23 July 2004; pp. 815–818.
38. Basseville, M.; Nikiforov, I.V. *Detection of Abrupt Changes: Theory and Application*; Prentice Hall Englewood Cliffs: New York, NY, USA, 1993; Volume 104.
39. Frías-Blanco, I.; Campo-Ávila, J.d.; Ramos-Jiménez, G.; Morales-Bueno, R.; Ortiz-Díaz, A.; Caballero-Mota, Y. Online and Non-Parametric Drift Detection Methods Based on Hoeffding’s Bounds. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 810–823. [\[CrossRef\]](#)
40. Nishida, K.; Yamauchi, K. Detecting concept drift using statistical testing. In Proceedings of the International Conference on Discovery Science, Sendai, Japan, 1–4 October 2007; pp. 264–269.
41. Raab, C.; Heusinger, M.; Schleif, F.-M.J.N. Reactive soft prototype computing for concept drift streams. *Neurocomputing* **2020**, *416*, 340–351. [\[CrossRef\]](#)
42. Berger, V.W.; Zhou, Y. Kolmogorov—Smirnov Test: Overview. 2014. Available online: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat06558> (accessed on 5 July 2022).
43. Gözüaık, Ö.; Can, F. Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artif. Intell. Rev.* **2020**, *54*, 3725–3747. [\[CrossRef\]](#)
44. Gonçalves, P.M., Jr.; de Carvalho Santos, S.G.; Barros, R.S.; Vieira, D.C.J.E.S.w.A. A comparative study on concept drift detectors. *Expert. Syst. Appl.* **2014**, *41*, 8144–8156. [\[CrossRef\]](#)
45. Jaramillo-Valbuena, S.; Londoño-Peláez, J.M.; Cardona, S.A. Performance evaluation of concept drift detection techniques in the presence of noise. *Perform. Eval.* **2017**, *38*, 16.
46. Barros, R.S.M.; Santos, S.G.T.C. A large-scale comparison of concept drift detectors. *Inf. Sci.* **2018**, *451–452*, 348–370. [\[CrossRef\]](#)
47. Liu, W.; Zhang, H.; Ding, Z.; Liu, Q.; Zhu, C. A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowl.-Based Syst.* **2021**, *215*, 106778. [\[CrossRef\]](#)
48. Zyblewski, P.; Sabourin, R.; Woźniak, M. Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. *Inf. Fusion* **2021**, *66*, 138–154. [\[CrossRef\]](#)
49. Bifet, A.; Holmes, G.; Pfahringer, B.; Kranen, P.; Kremer, H.; Jansen, T.; Seidl, T. Moa: Massive online analysis, a framework for stream classification and clustering. In Proceedings of the First Workshop on Applications of Pattern Analysis, Windsor, UK, 1–3 September 2010; pp. 44–50.
50. Hulten, G.; Spencer, L.; Domingos, P. Mining time-changing data streams. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; pp. 97–106. [\[CrossRef\]](#)
51. Ross, G.J.; Adams, N.M.; Tasoulis, D.K.; Hand, D.J. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recogn. Lett.* **2012**, *33*, 191–198. [\[CrossRef\]](#)
52. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [\[CrossRef\]](#)
53. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.