

CDDM: Concept Drift Detection Model for Data Stream

<https://doi.org/10.3991/ijim.v14i10.14803>

Mashail Shaeel Althabiti ^(✉), Manal Abdullah
King Abdulaziz University, Jeddah, Saudi Arabia
msalthubaiti@kau.edu.sa

Abstract—Data stream is the huge amount of data generated in various fields, including financial processes, social media activities, Internet of Things applications, and many others. Such data cannot be processed through traditional data mining algorithms due to several constraints, including limited memory, data speed, and dynamic environment. Concept Drift is known as the main constraint of data stream mining, mainly in the classification task. It refers to the change in the data stream underlying distribution over time. Thus, it results in accuracy deterioration of classification models and wrong predictions. Spam emails, consumer behavior changes, and adversary activates, are examples of Concept Drift. In this paper, a Concept Drift detection model is introduced, Concept Drift Detection Model (CDDM). It monitors the accuracy of the classification model over a sliding window, assuming the decline in accuracy indicates a drift occurrence. A modification over CDDM is a weighted version of the CDDM as W-CDDM.

Both models have evaluated against two real datasets and four artificial datasets. The experimental results of abrupt drift show that CDDM, W-CDDM outperforms the other models in the dataset of 100K and 1M instances, respectively. Regarding gradual drift, the W-CDDM overtook the rest in terms of accuracy, run time, and detection delays in the dataset of 100 K instances. While in the dataset of 1M instances, CDDM has got the highest accuracy using the NB classifier. Moreover, W-CDDM achieves the highest accuracy on real datasets.

Keywords—Data Stream Mining, Concept Drift, Concept Drift Detection, Data Stream Classification.

1 Introduction

Internet of things IoTs, weather forecasting, telecommunications systems, and many other applications are examples of data stream applications. The utilization of these types of applications has resulted in the vast open-ended stream of data. In any field, the generated data needs to be analyzed and, further, extract knowledge from the data repository for different purposes. Learning from such data differs from the static data sources, where it has some limitations, such as limited memory and a one-time data scan only [1]. One of the main constraints that causes the change of the data stream is the change in the underlying distribution, and accordingly, misclassification is the Concept Drift. Concept drift or drift, in general, is the phenomenon that deteriorates the accuracy of a learning model. It can be handled in the classification models through

different approaches, such as tracking the data probability distribution, controlling the model accuracy, or monitoring features changes [2].

In this paper, two concept drift detection models are introduced named, Concept Drift Detection Model CDDM and Weighted-CDDM W-CDDM. The proposed models will monitor the learning process and discover the changes in the underlying data stream distribution. Both are implemented in Java and tested using the MOA framework. The models are evaluated by comparing their performance against two other models from the literature, which are FHDDM and MDDM-A. The reason behind choosing these models is that they use single classifier and sliding window.

The rest of this paper is organized as follows. The general data stream classification model is described in Section 2. Section 3 presents the related works. The CDDM and W-CDDM are introduced in Section 4 and 5, respectively. The evaluations of the models against FHDDM and MDDM-A are presented in Section 6. The experimental results and the discussions are shown in Section 7 and 8.

2 General Data Stream Classification Model

The data stream classification model scans the incoming data stream only once in a dynamic environment to perform the classification task [3]. It tracks and adapts Concept Drift running in a limited amount of memory and time. After drift is detected, it incorporates the new concept by updating the classifier to maintain the model's accuracy and discards the old data. In a changing environment, a data stream classification model must consider concept drift to represent the real data situation with high prediction accuracy. As shown in Figure 1, the model should handle and adapt the concept drift while the learning process keeps progressing. Also, it should be able to distinguish the concept drift from noise and adapt the classification to the recent concept. Generally, the data stream classification algorithm has three functions: predict the class label of an instance, diagnose if concept drift occurred, and update the model [3] if required, which detailed in the following subsections.

2.1 Phase one: Classification

Classifying a streaming data is very different from static data in which training the model is done considering some constraints, including the limited memory space, the high speed of data arrival, and the one scan of instances. In this phase, the classifier identifies the class label of the instances. The incoming data stream \mathcal{S} is a sequence of instances that comes in the form of $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \dots$ represents the training data, which is utilized to build the classifier incrementally [4]. As instance x_t arrived, the classifier predicts its class label \hat{y}_t and updates the model accuracy. Then it accepts the next instance (x_{t+1}, y_{t+1}) to repeats the same process. The way of prediction is differed according to the type of classifier used.

2.2 Phase two: Concept drift diagnosis

The estimator plays an essential role in this phase, where it diagnoses the drift in the stream distribution. It works by monitoring the input stream to estimate desired statistics of important parameters [16]. Examples of stream estimators are estimating the mean, error rate, standard deviation, the weighted moving mean, entropy, or many others.

2.3 Concept drift detection

The primary function of classification is to decide the class label of a given instance, which can be described through the Bayesian Decision Theory in Equation 1 [4]:

$$\mathcal{P}(y_i | \mathcal{X}) = \frac{\mathcal{P}(y_i) * \mathcal{P}(\mathcal{X} | y_i)}{\mathcal{P}(\mathcal{X})} \quad (1)$$

Concept Drift occurs if the joint probability distribution at t_0 is not equal to the joint probability distributions at t_1 , which can be represented as follows:

$$\exists X = \mathcal{P}_{t_0}(\mathcal{X}, \mathcal{Y}) \neq \mathcal{P}_{t_1}(\mathcal{X}, \mathcal{Y}) \quad (2)$$

Concept Drift may take place in the stream due to the change in:

- The prior probabilities of classes $\mathcal{P}(y_i)$
- The class-conditional probability distributions $\mathcal{P}(\mathcal{X} | y_i)$, or
- The posterior probabilities $\mathcal{P}(y_i | \mathcal{X})$.

There are several patterns of Concept Drift such as :(1) abrupt drift; which occurs suddenly, (2) gradual drift; that emerges in un suddenly way over the time, and (3) reoccurring drift; that reappear after some time.

Concept Drift detection is an essential phase in the data stream classification model, where it detects Concept Drift. Many techniques monitor the classification error rate, or the distance between two errors includes DDM [5], and EDDM [6]. While, some other techniques monitor some performance indicators such as FHDDM [7].

The importance of this phase is to make the data stream classification model adaptive. It is responsible for signaling the drift alarm, determining which data to discard, and how to adapt the change. The detection methods in the literature are categorized into informed or blind methods [1]. The informed method detects the drift explicitly. Examples of such type are DDM [5], EDDM [6], ADWIN [8]. After the drift occurrence, it triggers the learner to be updated.

On the contrary, blind method handle Concept Drift by updating the learner regularly whether drift has occurred or not. Consequently, it leads to a high cost of resources and wasting time.

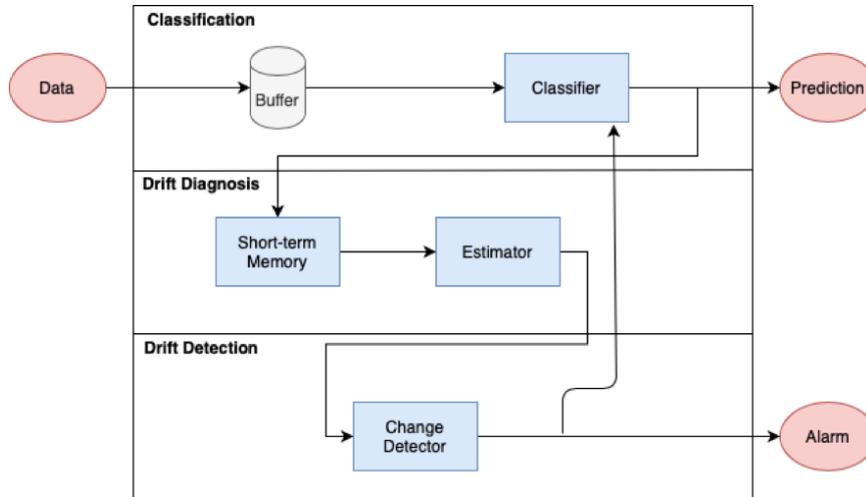


Fig. 1. General Classification Model with Concept Drift.

3 Related Work

A lot of researches have been developed to address the field of data stream classification with Concept Drift. *The Drift Detection Method* (DDM) [5] is one of the most well-known statistical-based detectors. It works based on the assumption that the increase in the model error rate suggests a drift. It monitors the model's error rate \mathcal{P} with its standard deviation S and two registers \mathcal{P}_{\min} and S_{\min} , which are updated when the $\mathcal{P}_t + S_t < \mathcal{P}_{\min} + S_{\min}$. DDM detects a drift when the $\mathcal{P}_t + S_t \geq \mathcal{P}_{\min} + 3 * S_{\min}$.

The Early Drift Detection Method (EDDM) [6] is a modified version of DDM, which assumes that the increase in the distance between two errors suggests a drift. It calculates the average distance between two recent errors \mathcal{P} and its standard deviation S . Also, it manages two registers \mathcal{P}_{\max} and S_{\max} , which are updated when the $\mathcal{P}_t + 2 * S_t > \mathcal{P}_{\max} + 2 * S_{\max}$, EDDM detects a drift when the $(\mathcal{P}_t + 2 * S_t) / (\mathcal{P}_{\max} + 2 * S_{\max}) > 0.90$. *Reactive Drift Detection Method* (RDDM) [9] is developed to improve the DDM performance. The proposed algorithm has overcome the problem of performance loss of DDM by discarding the older examples. It periodically recalculates the DMM calculations that determine the alarm and drift levels. Also, the drift occurs whenever the number of examples in the alarm level reaches the threshold.

The Fast Hoeffding Drift Detection Method (FHDDM) is a window-based method [7]. It slides a window which filled with the classifier's prediction results. It calculates the probability of correct classification \mathcal{P} of the data in the window at time t and store the highest value reached so far in the \mathcal{P}_{\max} variable. FHDDM detects a drift when $\mathcal{P}_{\max} - \mathcal{P} \geq \text{epsilon}$. Hoeffding's inequality is employed to calculate the value of the parameter epsilon.

Similar to FHDDM, *McDiarmid Drift Detection Method* (MDDM-A) [4] employs a sliding window, which filled with the classifier's predictions [4]. It calculates the

arithmetic weighted mean of the data in the window μ_w^t and keeps the maximum mean observed so far μ_w^{\max} . It uses McDiarmid's inequality to determine the drift epsilon. MDDM-A detects a drift if $\mu_w^{\max} - \mu_w^t \geq \text{epsilon}$.

Also, *The Adaptive Windowing Algorithm* (ADWIN) [8] slides a window of variable size that shrinks if drift is detected; otherwise, it keeps enlarging. It maintains two adjacent sub-windows; one represents the recent data and the other one for old data. Drift is detected if the different means of the sub-windows exceed the drift threshold.

In the paper, the developed CDDM and W-CDDM are introduced to maintain the classification model with high accuracy. The main goal is to get the least False Positive as it is one of the shortcomings of the previously mentioned methods. Furthermore, these model's applicability will help in avoiding the wastage of resources and minimize in the time involved due to false drift alarms. Let us consider we have a monitoring system; the high false alarms would result in time and resources wastage due to the unneeded maintenance. Hence, while developing the proposed model in the present research endeavor, the researcher will also focus on eliminating such fault lines.

4 Concept Drift Detection Model (CDDM)

The CDDM monitors the classification accuracy to track Concept Drift by estimating the probability of correct classification and its standard deviation. Figure 2 shows CDDM architecture and workflow. CDDM is designed with a generic sliding window of size n , considering the recent data is informative. The window will be filled with the results of the classifier's predictions, such that the algorithm inserts one if the prediction is correct, and zero if it is not, stored on the basis of first-in-first-out. As inputs processed, the estimator monitors the results of the classification in the sliding window during the learning. It calculates the P_SD which, represents the probability of correct classification \mathcal{P}_w and its standard deviation σ_w , computed over the window at time t .

Also, it keeps the highest result of the calculation observed so far in P_SD_{max} , which its value updated when the P_SD at time t is higher than the value of P_SD_{max} as shown in Equation 3.

$$\text{if } P_SD_{max} < P_SD_t \Rightarrow P_SD_t \rightarrow P_SD_{max} \quad (3)$$

Where

$$P_SD_t = \mathcal{P}_w + \sigma_w \quad \text{at time } t$$

According to probably approximately correct (PAC) learning model [5], as the number of instances increases, the classification accuracy would increase or stay steady. Accuracy degradation indicates the possibility of finding Concept Drift. So, probability of observing correct classification \mathcal{P}_w should increase or remain steady as instances processed. Otherwise, a Concept Drift in the evolving stream may occur.

Hoeffding's inequality [7] is one of the main probability inequalities used in data stream mining studies to bound the differences between the averages of the data stream. It has been used to define the epsilon of several detection methods such as FHDDM

[7], $HDDM_{A-tets}$ [10], and $HDDM_{w-tets}$ [10]. In probability theory and machine learning, it is used to identify the upper bound of the difference between the expected mean and the actual mean. In data stream studies, given the δ , which is the probability of error allowed, the Hoeffding's inequality guarantees that a drift has happened if the difference between $P_{SD_{max}}$ and P_{SD_t} exceeds the Epsilon as shown in Equation 4. Also, the Epsilon calculation is shown in Equation 5.

$$P_{SD_{max}} - P_{SD_t} > \epsilon \Rightarrow Drift := True \tag{4}$$

$$Epsilon (\epsilon) = \sqrt{\frac{1}{2n} \ln \frac{1}{\delta}} \tag{5}$$

Where n is the window size, and δ is the probability of error allowed which is a given value. A lot of experiments have been conducted to choose the suitable delta value for CDDM's Drift Epsilon, which is equal to 10^{-6} .

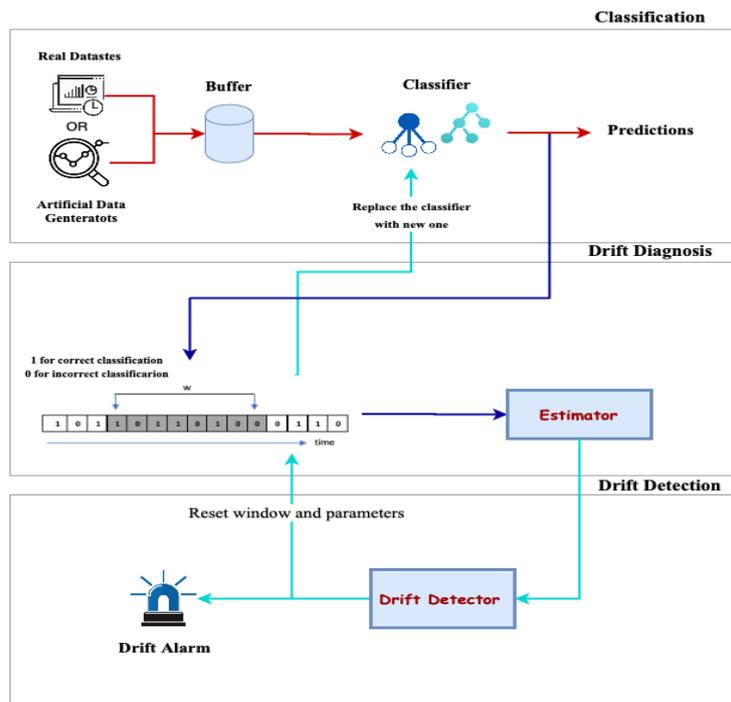


Fig. 2. The Concept Drift Detection Model CDDM.

Figure 3 illustrates an example of how the estimator tracks the Concept Drift. A sliding window of size six is filled with the results of the classification. The probability of correct classification and its standard deviation are computed while keeping the maximum value of it obtained so far. The difference between them will be compared to a threshold, which is equal to 0.36 in this example. If the difference between them exceeds a threshold, then a drift has been occurred.

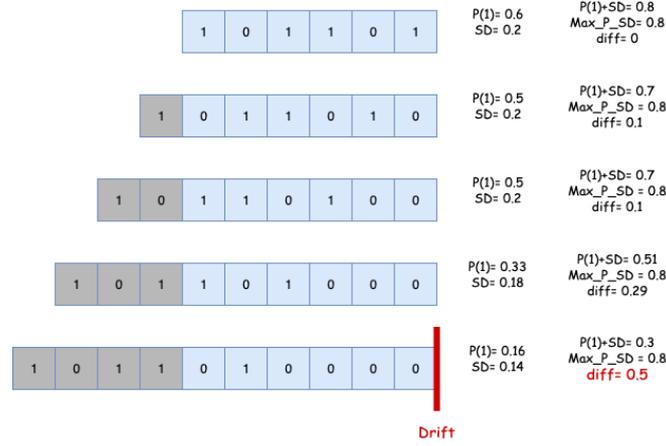


Fig. 3. The Estimation Process

CDDM falls under an informed detection method, which detects the drift explicitly to save time and resources. It monitors the estimations produced by the estimator and signals a drift alarm if the condition in Equation 4 is met. After the drift occurrence, it triggers the learner to be replaced with a new one. Also, it resets the sliding window and the parameters.

5 Weighted Concept Drift Detection Model W-CDDM

The CDDM can be improved by assigning weights to the probability of correct classification $\mathcal{P}(1)$ and its standard deviation σ , indicating its importance and contribution in computing the value of the P_SD . In order to choose the best weights values, various experiments with different values have been run. W-CDDM is a weighted version of the CDDM that assigns a weight of 0.8 to the probability of correct classification and 0.2 to its standard deviation, as shown in Equation 6. Also, it uses a fixed threshold of 0.2 instead of the epsilon computed by the Hoeffding inequality.

$$P_SD_t = (\mathcal{P}_w * 0.8) + (\sigma_w * 0.2) \quad (6)$$

6 Models Evaluation

In this section, the performance evaluation of the CDDM and W-CDDM in detecting abrupt and gradual drift is detailed.

6.1 Datasets

In the field of data stream classification, several datasets have been introduced to simulate the data in the non-stationary distribution. There are two types of datasets: Artificial and Real datasets. Real datasets are used to represent real-world applications, just a few of them are suitable for evaluating classification algorithms in a non-stationary environment [11]. They either do not have sufficient instances or do not contain any Concept Drift. Also, one of its disadvantages is the disability to determine the position of the drift [12]. Thus, it is hard to evaluate the detection delay, True Positive (TP), False Positive (FP), and False Negative (FN). Consequently, artificial generators have been proposed. In such a generator, drift's position and drift's length can be determined. Both artificial and real datasets will be used to assess the performance of CDDM and W-CDDM. Examples for both cases are:

Airline (Real): It is one of the popular real datasets used in evaluating data stream mining specifically for classification purposes [13]. It contains 539,384 instances with seven attributes. This dataset used to predict whether a given flight will be delayed or not.

Forest Coverttype (Real): It is a multivariate dataset with 581, 012 instances, and 54 attributes mainly used for classification problems [14]. It is used to predict the type of forest cover of wilderness areas located in Colorado.

AGRAWAL (Artificial): It is used to generate a data stream of people who intend to receive a loan [9]. It has nine attributes. To perform the classification, the authors have used the attributes to develop ten functions. Concept Drift is simulated by changing the classification functions.

SEA (Artificial): It is one of the well-known stream generators employed to evaluate data stream mining with concept drift detection algorithms. It has four classification functions, each with three attributes where only two of them are relevant to classification [15]. Concept Drift appears as a result of changing the classification functions.

6.2 Evaluation measures

On the evaluation of the performance of the CDDM and W-CDDM models, two perspectives are considered, the evaluation of the classification process and the concept drift detection process. Accuracy score and run time are well-known evaluation measures for classification algorithms. The accuracy score evaluates the overall model performance. It is calculated by dividing the number of correctly predicted instances by the total classifier's predictions [16]. The Run Time (CPU-seconds) is the total runtime involved for training and evaluating a data stream model [17].

Besides, additional measures will be applied to evaluate the models' performance in detecting the Concept Drift. A data stream classification model with a drift detector should achieve a high true positive (TP), low false positive (FP), and low false negative (FN), which can be computed on artificial datasets only, where drift position is known in advance [7]. The Acceptable delay length is a measure developed to measure the TP, FP, and FN of a data stream classification model. It mainly sets a threshold Δ to determine the acceptable range, which starts from the correct position of Concept Drift. Any

drift that occurs in this range is considered a true positive. First, the values of TP, FP, and FN are initialized to zeros. Then, when a Concept Drift is detected, there are values updated as follows:

- The True Positive is increased by one if the model is able to truly discover that a drift occurred at time t within the acceptable delay range $[t-\Delta, t+\Delta]$.
- The False Negative is increased by one when the model overlooks a correct drift that occurred at time t within the acceptable delay range $[t-\Delta, t+\Delta]$.
- The False Positive is increased by one when the model incorrectly detects a drift located outside the of acceptable detection intervals.

Moreover, the time elapsed before detecting Concept Drift can be determined, which is referred to as the drift detection delay [3]. It computes the time between real drift's occurrence and its detection.

6.3 Experimental setting

The proposed models CDDM and W-CDDM are developed in JAVA. In order to evaluate them, the MOA framework is used to carry out the experiments. MOA comes with the most common drift detection algorithms and provides a wide range of artificial generators. CDDM and W-CDDM will be compared with MDDM-A, FHDDM. The reasons behind choosing these two models are that they use single classifier and sliding window. Following the studies in the literature, they will be run using two incremental classifiers, Naive Bayes and Hoeffding Tree. All experiments were executed using a MacBook Pro with a 2.5 GHz Dual-Core Intel Core i5, 4 GB 1600 MHz DDR3 Memory (ram), and MacOS Catalina.

As mentioned before, two real datasets are used in this research, Forest Covertype, and Airline. We have employed a window size of 100 for all models as it results in better accuracy in the preliminary experiments.

Moreover, four artificial datasets have been generated using Agrawal and Sea generators. In the experiments with abrupt drift, the window size has been set to 30 for the dataset of 100k, 300 for the dataset of 1M, and the acceptable delay length is set to 250. While, in the experiments with gradual drift, the window size has been enlarged to 100 for the dataset of 100K, 1000 for the dataset of 1M, and the acceptable delay length to 1000. A wider window is mandatory to enable detect gradual drift. All the experiments setting is shown in Table 1. Also, four scenarios have been introduced; the first and second scenarios simulate the abrupt drift. The third and fourth scenarios simulate the gradual drift. Each scenario has different dataset sizes as listed below:

Scenario 1: A data stream of 100,000 instances was generated from the Agrawal data stream generator. The concept drift type is abrupt, simulated to occur every 20,000 instances with a transition length of 50 from a concept to others.

Scenario 2: A data stream of 1,000,000 instances was generated from the Agrawal data stream generator. The concept drift type is abrupt, simulated to occur every 200,000 instances with a transition length of 50 from a concept to others.

-

Scenario 3: A data stream of 100,000 instances was generated from the Sea data stream generator. The concept drift type gradual, simulated to occur every 25,000 instances with a transition length of 500 from a concept to others.

•

Scenario 4: A data stream of 1,000,000 instances was generated from the Sea data stream generator. The concept drift type is gradual, simulated to occur every 250,000 instances with a transition length of 500 from a concept to others.

•

Table 1. The Models' Setting.

Experiments	The Model Setting		
	Window size	Delta	Number of instances
Real dataset		10 ⁻⁶ .	Airline=539,384 Covertime= 581, 012
First Scenario (Abrupt Drift)	30		100,000
Second Scenario (Abrupt Drift)	300		1,000,000
Third Scenario (Gradual Drift)	100		100,000
Forth Scenario (Gradual Drift)	1000		1,000,000

A *prequential* evaluation is implemented to evaluate the CDDM and W-CDDM models. It has been designed to evaluate learning models that evolve over time [3]. It works by using each incoming instance for both testing and training. It uses the instance to test the model before using it for training and then update the model's accuracy. Thus, the model's ability to predict instances that have never seen before is assessed.

7 Results

The results of all the experiments are carried out to evaluate the CDDM, and W-CDDM models, against (FHDDM, and MDDM-A) are presented in this section. The algorithms are listed according to the accuracy from the highest to lowest. In artificial datasets, the classification accuracy, and run times are averaged and presented. Besides, the detection delays, TF, FN, and FP of the concept drift detection are provided. It is important to note that all results are average of results run between 10-30 times. The tables show for each scenario with different datasets accuracy, time, TP, FP, FN and delay as discussed above. The *No_detection* entry in the following tables shows the accuracy of running the classifier with the same drifted-dataset without drift handling. Table 2 and Table 3 show the results of running the first and second scenario, respectively.

Table 2. The Results of the First Scenario

Dataset	Classifier	Models	Accuracy	Runtime	TP	FP	FN	Delay
AGRAWAL (100k)	NB	CDDM	76.28	0.58	3	5	1	0.17
		FHDDM	76.01	0.64	3	14	1	0.1
		MDDM-A	75.57	0.67	3	18	1	0.067
		W-CDDM	68.57	0.66	2	440	2	0.15
		No-Detection	64.16					
	HT	CDDM	83.86	1.03	3	3	1	0.78
		FHDDM	83.5	1.16	3	13	1	0.73
		MDDM-A	80.29	1.18	3	20	1	0.29
		W-CDDM	70.11	1.56	3	401	1	0.091
		No-Detection	77.29					

Table 3. The Results of the Second Scenario

Dataset	Classifier	Models	Accuracy	Runtime	TP	FP	FN	Delay
AGRAWAL (1M)	NB	W-CDDM	76.32	2.19	3	0	1	0.007
		CDDM	76.16	2.86	3	25	1	0.011
		MDDM-A	76.16	2.93	3	45	1	0.006
		FHDDM	76.15	2.4	3	31	1	0.005
		No-Detection	63.31					
	HT	W-CDDM	88.62	4.7	3	0	1	0.35
		FHDDM	88.23	4.31	3	9	1	0.11
		MDDM-A	88.2	4.4	3	12	1	0.1
		CDDM	87.42	4.3	3	7	1	0.088
		No-Detection	78.45					

Table 4 shows the results of running scenario three, which simulates gradual drift with 100,000 instances. Another experiment simulates the same type of drift with one million instances being run and the results are presented in Table 5.

Table 4. The Results of the Third Scenario

Dataset	Classifier	Models	Accuracy	Runtime	TP	FP	FN	Delay
SEA (100k)	NB	W-CDDM	87.84	0.42	2	1	1	0.58
		MDDM-A	87.52	0.47	2	0	1	0.69
		FHDDM	86.9	0.45	1	1	2	0.27
		CDDM	86.6	0.39	0	1	3	0
		No-Detection	85.34					
	HT	FHDDM	86.53	0.74	0	1	3	0
		CDDM	86.43	0.71	0	1	3	0
		W-CDDM	86.13	0.67	1	4	2	0.27
		MDDM-A	86.13	0.89	1	4	2	1.35
		No-Detection	85.56					

Table 5. The Results of the Fourth Scenario

Dataset	Classifier	Models	Accuracy	Runtime	TP	FP	FN	Delay
SEA (1M)	NB	CDDM	88.25	2.08	2	1	1	0.012
		MDDM-A	88.25	3.75	2	1	1	0.014
		FHDDM	88.24	2.1	2	1	1	0.11
		W-CDDM	86.11	0	0	0	3	0
		No-Detection	86.11					
	HT	FHDDM	88.91	3.35	3	0	0	0.31
		CDDM	88.9	3.55	3	0	0	0.293
		MDDM-A	88.89	4.66	3	0	0	0.295
		W-CDDM	86.93	3.52	0	0	3	0
		No-Detection	86.93					

As the Concept Drift location is not known in real-world datasets, evaluating the detection performance is not possible. The results of evaluating the models on the Airline dataset and Forest Covertype with NB and HT are presented in Table 6. Also, the number of Concept Drift discovered are shown.

Table 6. The Results of the Real Datasets

Dataset	Models	NB			Models	HT		
		Accuracy	Run time	# of drift		Accuracy	Run time	# of drift
Airline	CDDM	67.38	2.66	116	W-CDDM	66.35	10.23	200
	W-CDDM	67.13	2.74	151	MDDM-A	66.34	10.41	188
	FHDDM	67.06	2.61	124	FHDDM	66.04	11.01	169
	MDDM-A	66.92	2.76	151	CDDM	65.87	10.53	162
	No_Detection	64.32			No_Detection	65.04		
Cov-ertype	W-CDDM	84.97	7.81	1206	W-CDDM	85.07	14.23	1128
	MDDM-A	84.89	7.94	1319	MDDM-A	84.98	14.53	1230
	FHDDM	84.82	7.59	1172	FHDDM	84.79	14.02	1110
	CDDM	84.68	7.76	1001	CDDM	84.71	13.47	931
	No_Detection	60.00			No_Detection	79.98		

8 Discussions

8.1 Regarding artificial datasets

The CDDM and W-CDDM are evaluated in detecting abrupt drift through scenarios one and two. The results of the first scenario with NB and HT are summarized in Table 2, which shows that CDDM is the highest in term of accuracy, shortest run time, and has the lowest value of FP among other models using both classifiers. This means that CDDM can detect exact drifts and not exaggerate the existing drifts in the dataset. From the table, we can also notice that W-CDDM has the least accuracy by 68.57 with NB and 70.11% with HT and the longest run time. The reason is due to the high FP value,

which indicates that W-CDDM is falsely considering valid instances as drift. It is worth noting that this does not mean that the model is not valid for drift detection because adjusting window size can reduce the FP and FN values. However, the results of enlarging the window size to 150 for all models show that W-CDDM’s FP’s value is reduced to three. One of the most important performances in stream mining is the detection delay where MDDM-A results in the shortest delay in NB experiment with 0.067 seconds, while the W-CDDM is the shortest in the HT experiment with 0.091 seconds. The *No_detection* values in the experiment are 64.16% and 77.29 with NB and HT classifiers increased as a result of drift handling. Figure 4 shows the models’ accuracy through the classification process, where the dotted lines represent the drifts.

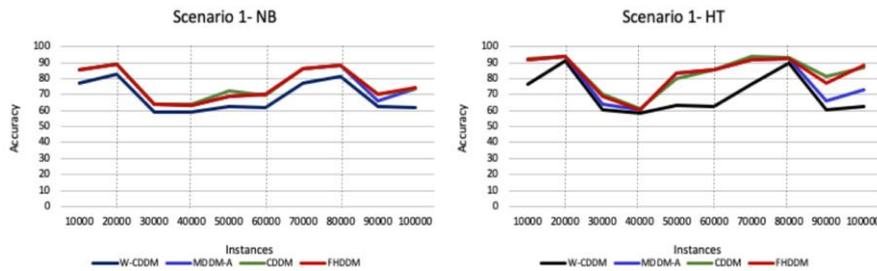


Fig. 4. The Accuracy Evolution of the First Scenario

From Figure 4, we can see that there are two possible cases after drift occurrence, an increase or decrease of accuracy. For example, After the drift in instance# 20,000, the accuracy has been decreased by instance# 30,000. This is means that the classifier takes time to recognize the drift and repair itself. In contrast, the accuracy has been increased in instance# 70,000 as a result of the drift in instance# 60,000. This indicates that the classifier has been rebuilt with new data after the detection.

The number of instances has increased in the second scenario to one million instances. This experiment aims to test the effect of dataset size on models’ performance. The results in Table 3 show that W-CDDM achieves the highest accuracy and lowest FP using both NB and HT classifiers. Also, it has the shortest classification run time of 2.19 s with the NB classifier, while CDDM run time is the shortest of 4.3 s using HT classifier. FHDDM results in the shortest detection delay of 0.005 s using NB, while CDDM detection delay is the best with 0.088 s in HT experiments. Figure 5 presents the models’ accuracy as instances increasing.

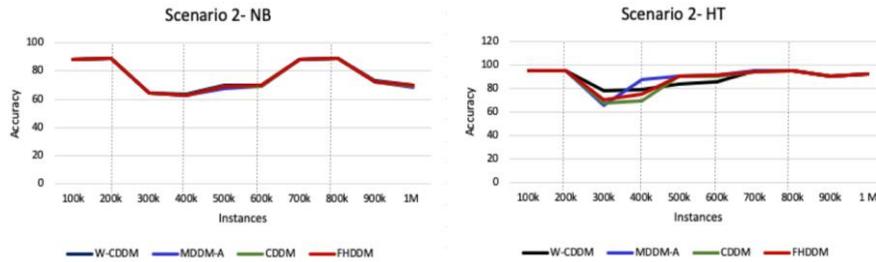


Fig. 5. The Accuracy Evolution of the Second Scenario

From Figure 5, we note that with NB classifier, all models have the same performance during the process, so their accuracy values are all relatively similar. In the HT classifier, the models' performance in first drift are varying, but then in the end data, their performance becomes similar.

The model's ability to detect gradual drift has been assessed in the third scenario. The results in Table 4 show that W-CDDM outperforms the rest in terms of accuracy, and detection delays, except in NB classifier experiments, the MDDM results in the lowest false positive. The result of running the same scenario with the HT classifier shows that FHDDM and CDDM have high FN value of 3 as they do not detect any drift; nonetheless, it has the highest accuracy. FHDDM has the lowest detection delay of 0.27 s; however, the reason for this is that only one drift is detected. While the other models such as MDDM_A and W-CDDM have detected two drifts, thus their delays are longer since it is the total of drifts delays. W-CDDM and MDDM have the same accuracy, but W-CDDM is better in the run time and detection delay. The accuracy evolutions of all the models using both classifiers are shown in Figure 6.

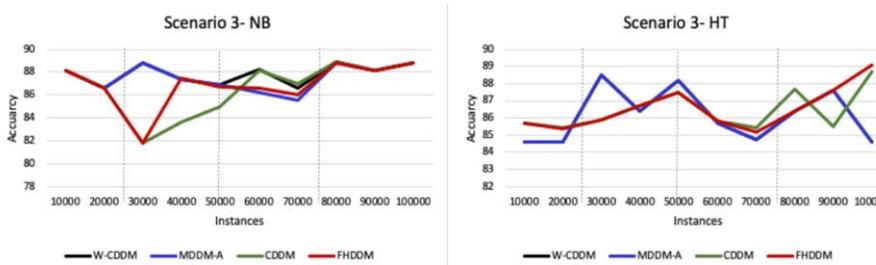


Fig. 6. The Accuracy Evolution of the Third Scenario

From Figure 6, we can observe the reason for the high accuracy of FHDDM and CDDM because they falsely considered drift in the end of instances, and accordingly, the classifier has been replaced with a new one using a new concept, so their accuracy increased.

Besides, CDDM and MDDM-A get the same accuracy value as do the TP, FP, and FN in the fourth scenario with the NB classifier, as shown in Table 5. Compared to MDDM-A, CDDM is better where it has the lowest run time of 2.08 s and detection

delay of 0.012 s. For HT, the FHDDM results in the highest accuracy of 88.91 %, followed by CDDM with 88.9 %. Also, CDDM results in the lowest detection delay of 0.293 s compared to the rest. Moreover, it noticed that the No_detection entry values are the same as the of W-CDDM's accuracy as it did not detect any drift. Consequently, it proves that detecting drift may improve the accuracy. Figure 7 shows the models' accuracy through the classification process.

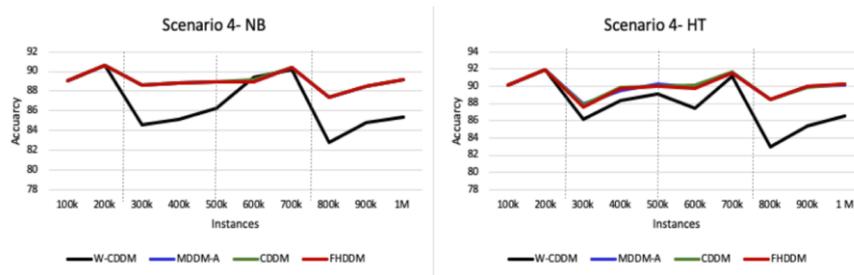


Fig. 7. The Accuracy Evolution of the Fourth Scenario

Figure 7 shows that W-CDDM accuracy is the lowest in all the instances stages where it does not detect any drift.

8.2 Regarding real datasets

The experimental results for Airline and Covertypes dataset in Table 6 show that the accuracies of the classification models have been increased as a result of employing concept drift detection. In the airline dataset, The CDDM and W-CDDM achieve the highest accuracy with NB classifier. In terms of classification run time, FHDDM results in the shortest time of 61s, followed by CDDM and W-CDDM. Also, the W-CDDM overtakes the rest of the models with a HT classifier in terms of accuracy and run time.

In the Covertypes dataset, the W-CDDM is the highest in the classification accuracy using both classifiers. Similar to the Airline dataset, FHDDM results in the shortest run time of 7.59 s using the NB classifier while CDDM is the shortest in classification runtime with 14.02 s using HT classifier.

9 Conclusion

In this paper, two models for concept drift detection in data streams are proposed, named CDDM and W-CDDM. They work by monitoring the classifier's predictions, which are filled in sliding a window of size n . Two variables are maintained through the learning process P_{SD_t} and $P_{SD_{max}}$, which indicate the probability of correct classification and its standard deviation. The drift is detected if the difference between these two variables exceeds the epsilon.

CDDM and W-CDDM have evaluated against FHDDM and MDDM-A through experiments were carried out using four artificial dataset simulating abrupt and gradual

concept drifts with different sizes, as well as two real-world datasets. The results of detecting abrupt drift on the artificial dataset with 100 K instances, show that CDDM outperformed existing models in detecting abrupt drift in terms of classification accuracy values, run time, and false positive. Also, W-CDDM was the best in accuracy and had the lowest FP value with both classifiers when the dataset has been increased to 1M instances. Regarding gradual drift, the results of evaluating the models on the dataset of 100 K instances showed that W-CDDM outperformed the rest in terms of accuracy, run time, and detection delays. While in the dataset with 1M instances, CDDM has got the highest accuracy using NB classifier and FHDDM is the highest in accuracy using HT classifier, followed by CDDM. Moreover, the W-CDDM achieves the highest accuracy on real datasets.

10 References

- [1] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, “Discussion and review on evolving data streams and concept drift adapting,” *Evol. Syst.*, vol. 9, no. 1, pp. 1–23, 2018. <https://doi.org/10.1007/s12530-016-9168-2>
- [2] Y. Kadwe and V. Suryawanshi, “A Review on Concept Drift,” *IOSR J. Comput. Eng.*, vol. 17, no. 1, pp. 20–26, 2015.
- [3] J. Gama, I. Zliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A Survey on Concept Drift Adaptation,” *ACM Comput. Surv.* 1, 1, Artic., vol. 1, no. 1, 2013. <https://doi.org/10.1145/2523813>
- [4] A. Pesaranghader, H. L. Viktor, and E. Paquet, “McDiarmid Drift Detection Methods for Evolving Data Streams,” in *Proceedings of the International Joint Conference on Neural Networks*, 2018. <https://doi.org/10.1109/ijcnn.2018.8489260>
- [5] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, “Learning with Drift Detection,” in *17th Brazilian Symposium on Artificial Intelligence Proc*, 2004, pp. 286–295. https://doi.org/10.1007/978-3-540-28645-5_29
- [6] M. Baenagarc et al., “Early drift detection method,” *Fourth Int. Work. Knowl. Discov. from data streams*, vol. 6, no. August 2014, pp. 77–86, 2006.
- [7] A. Pesaranghader and H. Viktor, “Fast Hoeffding Drift Detection Method for Evolving Data Streams,” in *Machine Learning and Knowledge Discovery in Data-bases*, 2016, pp. 96–111. https://doi.org/10.1007/978-3-319-46227-1_7
- [8] A. Bifet and R. Gavaldà, “Learning from Time-Changing Data with Adaptive Windowing,” pp. 443–448, 2013.
- [9] R. S. M. Barros, D. R. L. Cabral, P. M. Gonçalves, and S. G. T. C. Santos, “RDDM: Reactive drift detection method,” *Expert Syst. Appl.*, vol. 90, pp. 344–355, 2017. <https://doi.org/10.1016/j.eswa.2017.08.023>
- [10] C.- Avila and R. Moralesbueno, “Online and Non-Parametric Drift Detection Methods Based on Hoeffding’s Bounds,” vol. 27, no. 3, pp. 810–823, 2015.
- [11] D. Brzezinski and J. Stefanowski, “Reacting to Different Types of Concept Drift,” vol. 25, no. 1, pp. 81–94, 2014.
- [12] A. Bifet, “MOA: Massive Online Analysis Learning Examples,” *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010.
- [13] “Datasets.” [Online]. Available: <https://moa.cms.waikato.ac.nz/datasets/>. [Accessed: 20-Dec-2019].

- [14] “Covertime Data Set.” [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/covertime>. [Accessed: 28-Nov-2018].
- [15] A. Liu, G. Zhang, and J. Lu, “Fuzzy time windowing for gradual concept drift adaptation,” IEEE Int. Conf. Fuzzy Syst., 2017. <https://doi.org/10.1109/fuzz-ieee.2017.8015596>
- [16] A. Bifet, H. Geoff, and R. Kirkby, DATA STREAM MINING: A Practical Approach, no. May. 2011.
- [17] P. Dhaliwal and M. P. S. Bhatia, “Effective Handling of Recurring Concept Drifts in Data Streams,” Indian J. Sci. Technol., vol. 10, no. 30, pp. 1–6, 2017.

11 Authors

M.Shaeel Althabiti completed her master’s degree in information system Department at the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia in 2020. Her research field’s interest includes Data Science and Machine Learning.

M.Abdulaziz Abdullah. received her PhD in computers and systems engineering, Faculty of engineering, Ain Shams University, Cairo, Egypt, 2002. She has experienced in industrial computer networks and embedded systems. Her research interests include Artificial Intelligence, performance evaluation, WSN, network management, Big Data analysis, and pattern recognition. Dr Abdullah published more than 120 research papers in various international journals and conferences. She has also joined many HiCi research projects all over the world.

Article submitted 2020-04-13. Resubmitted 2020-05-04. Final acceptance 2020-05-05. Final version published as submitted by the authors.