Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 799–809

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Remediating data drifts and re-establishing ML models

Sreeja Ashok[a] *, Sangeetha Ezhumalai[b], Tanvi Patwa[c]

[a,b,c] *Analytics and Insights, Tata Consultancy Services, India*

Abstract

Machine learning models are significantly used in organizations for critical business processes and decision-making. The inputs and predictions from these models might drift over time due to several reasons, causing unanticipated behaviour and performance issues. COVID-19 has a significant impact on business resulting in huge dispersions of the data with unstable characteristics and irregular spikes at various time steps. When the data set experiences significant drifts due to the changes in their statistical properties, continuing with existing models based on historical data will lead to poor decision outcomes. It's hard to anticipate all significant future events and create an ML model which can withstand black swan events. Therefore, it's imperative to have an automated process for continuous monitoring in the ML lifecycle to identify the potential issues using drift detection methodologies with proper actions to avoid false predictions in silence. This study is focused on time series where data stream is represented as observations at regular intervals. Many dynamic real-world processes can be modelled using time series where data can flow quickly and change over time which is detrimental to time series analysis and forecasting. The objective of this study is to compare and evaluate different approaches for data drifts in time series models to retain its optimum performance and proposing an integrated approach for handling those drifts in real-time pipelines using timely alerts and re-establishing models.

© 2023 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering

Keywords: Data drifts, time series, statistical methods, sequential methods, window-based, drift detection

1. Introduction

Recently, many business processes are incorporating machine learning into their operations. The power of a model is its ability to discover relationships between a set of independent variables and dependent targets. A model will perform well on new data if the new data is consistent with the data it was trained on. Any machine learning model

* Corresponding author. Tel: +91- 9946044700.
E-mail address: sreeja.ashok@gmail.com

1877-0509 © 2023 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering
10.1016/j.procs.2023.01.060

works on the assumption that the data and logic it uses are accurate representations of reality. Unfortunately, the real world tends to change so much. Even if we train a model on an acceptable dataset, the distribution of incoming data will most likely shift over time. Many such changes can come from a wide range of sources. Market changes, purchase behaviour changes, or any external element can affect the underlying data acquired. This multitude of sources of change can significantly impact how model works, leading to model performance problems. Data drift, the primary contributors to this performance problem can be described as a disparity between the data used to test and evaluate the model before it was deployed in production and the real time data in production. It's critical to create a consistent procedure for detecting data drift and configuring proactive alerts to maintain the model quality over time. Analysing the model performance would also require extensive knowledge of the distribution and characteristics of the information, the kind of changes, and the use cases implemented. Drift detection approaches are used to characterize the inherent state of the system by detecting the times related to changes in the patterns and characteristics of time series data. Before attempting to identify drift using various methods, it is necessary to first understand various types of drifts that might occur and the factors causing them.

Drifts can be mainly classified as follows.

1. Data Drift/ Feature Drift: Occurs when the features of the independent variables used as input to the model changes.
2. Concept Drift: Occurs when there are changes in the properties of dependent variable, external factors can cause the label to evolve. Concept drift can manifest itself in a variety of ways. [1][2]
   Sudden drift – In a short timeframe, a transformation emerges. (e.g., COVID-19)
   Gradual & Incremental drift –Over the long term, new change gradually and progressively transforms an older one.
   Recurring concepts – After some time, an old occurrence of an event may resurface.

   Depending on the situation, concept drift can again be classified as virtual drift or real drift.

   2.1 Virtual Drift: When the independent variable changes but the predictions remain the same; here performance is not affected.
   2.2 Real Drift /Actual Drift: Here the performance of the model is affected. This can happen when the independent variable remains the same, but the decision boundary changes or when both the independent variable and the decision boundary changes.

Following are some of the factors that can cause drifts

- Upstream changes, such as the unit of input variable measurement changes.
- Data quality issues related to incomplete data or anomalies or unusual patterns.
- Natural changes due to weather conditions.
- Change in the relation between features.
- Model being trained on certain set of data but revealed to a larger range later.

The primary focus of this research is to compare and evaluate different drift detection methods for time series with varying distributions of data streams and to suggest a suitable workflow for real time analysis of time series models with prompt alerts and suitable actions to retain its performance. The paper is structured as follows. Different methods for handling drifts are explained in section 2 and section 3 details the pros and cons and drift detection criteria of each approach. Section 4 explain a process flow that helps in addressing the issues in real time and section 5 elaborates the results and discussions. Section 6 concludes with the final remarks and future research plan.

2. Approaches for detecting drifts

A wide range of metrics can allow us to identify and measure data drift with different functions and assumptions that are critical to each application. Several methods proposed in the literature for detecting drifts are statistical, statistical process control based, time window based, and sequential analysis based. Sequential analysis approaches rely on error rate to detect drifts whereas time window-based methods employ statistical distance computations to measure drifts between probability distributions. Statistical techniques that compute differences between adjacent segments of a time series are typically used to compute change points. Parametric or non-parametric techniques are commonly used to measure discrepancies between regions. The time intervals are represented by a probability density function (PDF) in parametric approaches, although such easy depictions restrict the kinds of changes that can be observed. Using kernel functions, non-parametric approaches provide more flexibility in representing the density functions of time intervals. Various methodologies explored by researchers and their ramifications are listed in this section.

2.1. Statistical based methods

2.1.1. Two sample Kolmogorov–Smirnov statistic (KS test)

KS test examines two cumulative distributions and provides the greatest difference. It's a non-parametric test, and thus doesn't require any test on data distribution assumptions. If the null hypothesis is rejected, the cumulative distributions will be different. Bonferroni correction can be used to minimize false positives by adjusting α values for different feature comparisons [3]. The Kolmogorov-Smirnov statistic, the computed p-value, is defined as follows:

$$KS_{p,q} = \max | F1(x) - F2\ (x)| \tag{1}$$

where p and q are the total observations from A and B; A and B are samples that include univariate observations.

2.1.2. Population Stability Index (PSI)

PSI measures how much a sample size has changed over time or between two independent samples of the same population [4]. It measures the variable shifts in distribution; it is symmetric and established using entropy measures [5][6]. PSI is widely used for accessing the stability of the model output and commonly used in banking sectors. Let N be the base population sample size and M be the target population sample size. PSI can be defined as

$$PSI = \sum_{I=1}^{B} \left(\frac{ni}{N} - \frac{mi}{M} \right) * ln\left(\frac{ni}{N} / \frac{mi}{M} \right) \tag{2}$$

where n_i and m_i are counts in the i^{th} bin, $\sum ni = N$ and $\sum mi = M$

2.1.3. Divergence scores from information theory

2.1.3.1. Kullback-Liebler divergence (KL)

KL divergence often known as relative entropy, is a measure of the variation between two probability distributions [7]. When one distribution has a high degree of variability or a limited sample size compared to the other, KL divergence is relevant. It is asymmetrical and the score will be always between 0 and infinity, with 0 indicating that the two distributions are similar. KLD is used to restrict the commitment of minority classes, which masks the genuine level of classifier incongruence and Iterative minimization of KLD is used for optimized image restoration and denoising [8]. KL divergence is calculated as a summation if P and Q represent discrete random distributions [9].

$$KL(p, q) = \sum p(x) \log\left(\frac{p(x)}{q(x)} \right) \tag{3}$$

2.1.3.2. Jensen-Shannon Divergence

Jensen-Shannon divergence is an extension of KL divergence that determines a symmetrical score and distance measure between two probability distributions.

$$JS\ (P, Q) = \sqrt{\frac{KL(\ P,M\ )+KL(\ Q,M\ )}{2}} \tag{4}$$

where M = (P + Q) / 2. When using log base 2, it returns scores ranging from 0 (similar distributions) to 1 (dissimilar distributions). JS divergence is utilized to develop a dissimilarity metric of data stream distribution, called Fault detection and classification (FDC) for runtime anomaly detection [10].

2.1.4. Wasserstein distance

Wasserstein metrics is a distance function initially used for optimal planning of transportation of goods and materials and later used in improving the framework of Generative Adversarial Networks (GAN) and for deriving practical formulas for several geometric quantities [11].

2.2. Time window-based approaches

2.2.1. Adaptive Windowing (ADWIN):

ADWIN, one of the most prominent algorithms for univariate drift monitoring is a robust sliding window approach for identifying shifts and maintaining data stream statistics. Rather than utilizing a fixed window size to identify drift, the approach uses a variable-size window to capture the data streams. By dividing the window at various locations and calculating the average of some statistics over these two windows, the algorithm will estimate the size of the window. When there is no variation in the window given, the window size is automatically determined while maintaining the key statistics for future data streams. Uncertainty Drift Detection (UDD) is an enhanced version of ADWIN for detecting structural changes over time and ADWIN2 is for performance optimization [12]-[14].

2.2.2. Hoeffding Drift Detection Method (HDDMS):

HDDM_A and HDDM_W are approaches for identifying abrupt and slow drifts respectively. To locate drifts, the first one measures moving averages and the second one uses exponentially weighted moving average (EMWA) for defining weights [15]. Hoeffding inequality is utilized in both circumstances to define an upper bound on the magnitude of difference between averages. FHHDM is an enhanced and fastest approach using sliding window and used in social network analysis [16][17].

2.3. Statistical Process Control based methods

2.3.1. Drift Detection Method (DDM)

DDM uses statistical process control (SPC) that keeps track of the base learner's error rate assumes that the failure rate will fall while the number of instances rises if the data distribution is stationary. It computes the minimum probability of error p_t and minimum standard deviation of error rate s_t at time t and updates values for p_{min} and s_{min} if $p_t+s_t < p_{min}+s_{min}$. DDM is good at detecting incremental (not extremely slow) and abrupt changes (sudden drifts). It has trouble identifying drifts that are modest and steady.

2.3.2. Early Drift Detection Method (EDDM)

Early Drift Detection Method (EDDM) is DDMs' enhanced version for spotting gradual drift [18]. It measures the average distance between two adjoining errors p_t and its standard deviation s_t at time t and updates values for p_{max} and s_{max} if $p_t+2s_t > p_{max} + 2\, s_{max}$. EDDM is more sensitive to noise while detecting changes than DDM. EDDM is used for analysing the classifier's accuracy using a distance measure rather than making any assumptions about feature distribution for detecting drifts [19]. The Reactive Drift Detection Method (RDDM) is a modified form of DDM that can discard outdated instances of very long concepts to discover drifts as soon as possible, it consumes more memory but has better accuracy than DDM [20][21].

2.4. Sequential analysis approaches

2.4.1. CUSUM \ Page-Hinckley (PH)

Page-Hinkley (PH) will identify a concept drift if the experimentally verified mean is larger than a predefined threshold value lambda [22]. Cumulative sums (CUSUM) are simple and efficient sequential technique that was originally created for process control but is now widely used to determine the underlying features of time series. Widely used applications of CUSUM are in environmental time series analysis for extracting driver-response relationships by plotting CUSUM-transformed variables against non-transformed variables [23]. A novel method which combines the CUSUM chart based on the generalized likelihood ratio with the exponentially weighted least square regression procedure is used for estimating shift size [24].

3. Advantages and disadvantages of drift detection approaches

A detailed evaluation of each approach used in this study is given below; Tables 1-4 summarizes the advantages and disadvantages of each approach; Table 5 represents drift detection measures or thresholds that can be applied for each approach.

Table 1: Pros and Cons of Statistical Analysis Methods

Approaches	Pros	Cons
Kolmogorov-Smirnov (KS) test	Non–parametric statistical test used on any data stream irrespective of the data distribution.	More likely to detect drift at the centre of the distribution than at the tails and the performance of detecting drifts at the tails is not good.
Population Stability Index (PSI)	Useful for datasets with stable distributions.	
Kullback-Leibler (KL) divergence/ Jensen-Shannon (JS) Divergence	Useful When the test dataset is small when compared to training dataset or the variance is high. JS divergence is symmetric.	KL divergence is not symmetric. For numerical features, discretization is required for both.

Table 2: Pros and Cons of Statistical Process Control Methods

Approaches	Pros	Cons
Drift Detection Method (DDM)/ Early Drift Detection Method (EDDM)/ Reactive Drift Detection Method (RDDM)	Memory consumption is low [30]. Performs better for abrupt changes [28]. Fast execution time [28]. For slow gradual changes, EDDM performs better than DDM [18] Low error rate [18] RDDM is optimized in terms of performance when compared to DDM by clearing earlier instances of data and thereby detecting drifts faster [15]	Performance of detecting slow gradual drifts is bad [28]. EDDM is more sensitive to noise than DDM [18].

Table 3: Pros and Cons of Sequential Analysis Methods

Approaches	Pros	Cons
Page Hinkley (PH)/ Cumulative Sum (CUSUM)	Memory consumption is low [28] CUSUM minimizes worst case delay [27]	Performance depends on the choice of parameters [28] CUSUM is effective only when the probability distributions of pre and post changes are known [27]

Table 4: Pros and Cons of Time Windowing Methods

Approaches	Pros	Cons
ADWIN (Adaptive Windowing)/ HDDM (Hoeffding Drift Detecting Method)	Configuring window sizes is not required. [25] Precisely locates the drift points [28] HDDMA−test is effective in detecting abrupt drifts and HDDMW−test is effective in detecting gradual drifts. [15]	Memory usage is high [30] Computationally expensive as it performs cut-check for all multiple sub windows. [25] Performance is low for multi-dimensional data since multiple windows are needed [26][31]

Table 5: Drift checks for each approach

Approaches	Drift detection
Kolmogorov-Smirnov (KS) test	p value > 0.05 - No drift (null hypothesis) ; p value < 0.05 - Drift occurred (alternative hypothesis)
Population Stability Index (PSI)	PSI < 0.1 - No drift; PSI >= 0.1 & PSI < 0.2 - slight change, good to analyse the changes. PSI >=0.2 - Drift occurred, model should be retrained or even remodelled.
Kullback-Leibler (KL) divergence	$0 \leq KL \leq$ infinity; 0 – same distributions; Lower the divergence, closer the distributions
Jensen-Shannon (JS) Divergence	$0 \leq JS \leq 1$; 0 – same distributions; 1 – Maximally Different distributions Lower the divergence, closer the distributions
Wasserstein distance	Lower the value, closer the streams
Drift Detection Method (DDM)	pi + si >= pmin + 2* smin -> Warning zone ; pi + si >= pmin + 2* smin -> Change detected pi : The error rate at instant i , si : The standard deviation at instant i , pmin : The minimum recorded error rate. smin : The minimum recorded standard deviation.
Early Drift Detection Method (EDDM)	(pi' + 2 * si')/ (pmax' + 2 * smax') < α -> Warning zone (pi' + 2 * si')/ (pmax' + 2 * smax') < β -> Change detected pi': running average distance at instant *i* ; si': running standard deviation at instant I pmax': Maximum recorded running average distance, smax': Maximum recorded running standard deviation, α and β are warning and out control level
PH/HDDM/ADWIN	Change detection & warning detection based on thresholds

4. Steps to deal with drifts

Introducing a process for drift detection in the life cycle of model management is vital to keep track of model performance over time and to ensure that the model continues to meet expectations. By setting up scheduled checks and customizing alert notifications the process can be monitored for big drops in KPIs used. An auto retraining process can be integrated into existing business process if significant variations are observed for each measure. Fig 1 details the process for spotting model degradation before it has a major impact on the business.

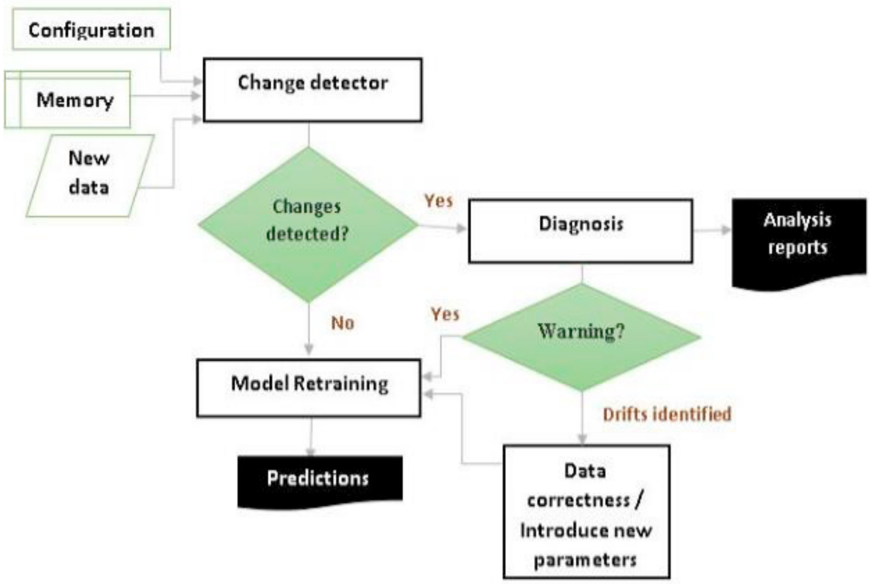

Fig. 1. Process Flow Diagram for Model Monitoring and Detecting Drifts

5. Results and discussion

The datasets and experimental parameters used in the drift detectors with evaluation techniques are outlined below.

5.1. Datasets

Experiments are conducted using three synthetic univariate time series datasets with varying drift patterns and the performance is compared using various evaluation metrics. Time series is a series of observations that have been captured in a precise order where observations are organized chronologically that demonstrate a serial correlation [32]. The shape of the series with distributions can be observed along with each pattern.

In the first dataset, we introduced a sudden change in the data stream. Fig.2. (a), (b) shows non–stationarity in the distribution due to sudden drifts. The size of data stream is 1500 out of which 500 data points fall in one distribution which is considered as training data and there is a sudden drift at time step 500, followed by a new behaviour and then recovered to a new normal stage with a different distribution.

In the second dataset, we established a recurrent drift. Fig.3 (a), (b) shows non–stationarity in data distribution due to recurrent drifts where old behaviour and new behaviour reoccur in a cyclic manner. The size of data stream is 2000 out of which first 500 points follow the old pattern, and the next 500 points follow new pattern recurrently.

In the third dataset, we introduced gradual drift where data slowly adapted to a new behaviour represented using Fig.4. (a),(b). The size of data stream is 999 and the drift started to appear gradually after 600 points.

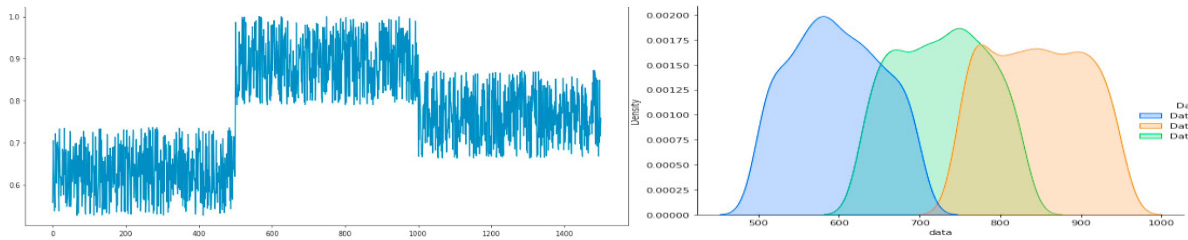

Fig. 2. Sudden drift (a) time series plot (b) distribution plot

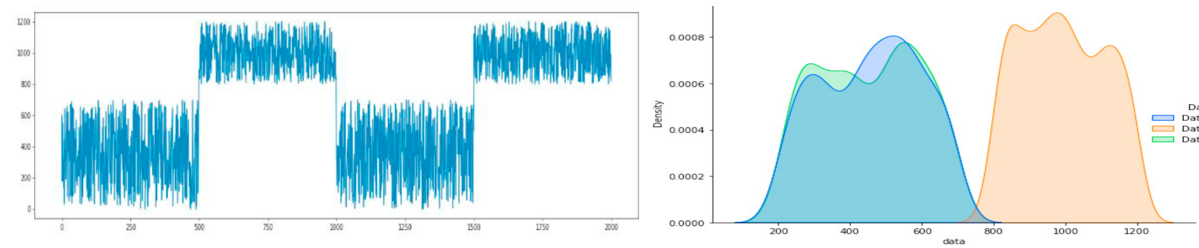

Fig. 3. Recurrent drift (a) time series plot (b) distribution plot

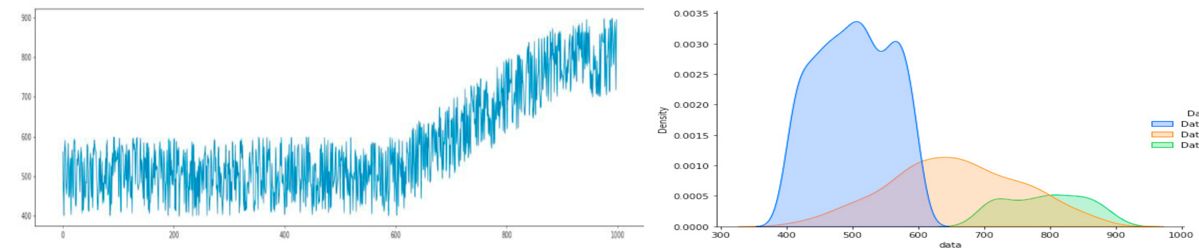

Fig. 4. Gradual drift (a) time series plot (b) distribution plot

5.2. Evaluation metrics

Different evaluation metrics for drift detection methods are detailed below in Table 6 [29].

Table 6: Evaluation metrics for Drift detection methods

Evaluation metrics	Definition
Detection delay	Performance indicator that gives the number of instances between the actual drift position and the one detected by the algorithm.
True Detection	Drift detected within the acceptable drift delay threshold
False Alarm	Drift detected outside the acceptable drift delay threshold
Detection run time	Time taken to detect drift
Memory Usage	Memory occupied by the algorithm
False positive rate	Gives the rate of wrongly detected drift when no actual drift occurs
False negative rate	Gives the rate of drift that are undetected.

If we take the following assumptions, the definitions of true/false positives and negatives are illustrated in Table 7.

- "Drift" as positive
- "No drift" as negative

Table 7: Description of true\false positives and negatives

True Positive (TP): Reality: Drift occurred. Outcome: Drift is detected.	False Positive (FP): Reality: No drift occurred. Outcome: Drift is detected.
False Negative (FN): Reality: Drift occurred Outcome: Drift is not detected.	True Negative (TN): Reality: No drift occurred. Outcome: Drift is not detected.

Usually, true negatives will not be counted because they will account for most of all data items (non-drift items). A tolerance threshold, € indicates that if the period between real drift starts and drift detection is less than €, we consider detected drift to be accurately identified. Drift detection approaches which are efficient should be more resilient towards false-positive rates while keeping the forecasting error rate low.

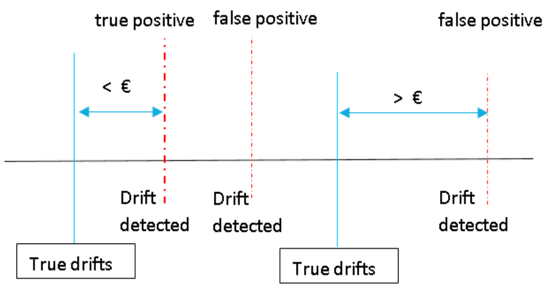

Fig. 5. Vertical lines specify the drifts detected.

5.2. Evaluation metrics

Real-time datasets usually follow huge variations in data distribution. To deal with varying distributions and to compare the performance, the analysis is done with normalization and without normalization. We used "scikit-multiflow" python package for exploring different drift detection approaches.

Table 8 shows the experimental results of statistical approaches against sudden drift, recurrent drift, and gradual drift. The results of both with\without normalization are similar for statistical approaches, though the magnitude of scores is different for few models.

Table 8: Experimental analysis using statistical approaches

Statistical Approaches	Without Normalization			With Normalization		
	Sudden drift	Recurrent drift	Gradual drift	Sudden drift	Recurrent drift	Gradual drift
KS test	7.39e-300	7.39e-300	3.32e-30	7.39e-300	7.39e-300	3.32e-30
PSI	6.91	6.91	0.56	6.91	6.91	0.56
KL divergence	-147186.60	-240529.40	-22944.63	-155.09	-200.77	-25.52
JS divergence	8589.66	47974.31	1283.62	9.05	40.04	1.42
Wasserstein	248.72	562.72	91.99	0.26	0.47	0.10

Figure 6 – 9 represents the visual outputs of a few algorithms based on different test scenarios.

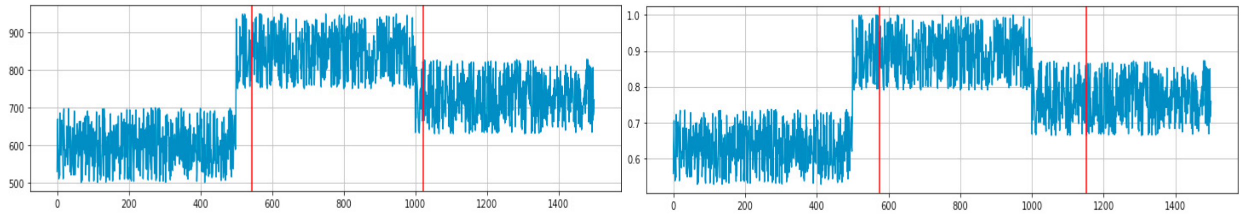

Fig. 6. ADWIN performance on sudden drift a) Without Normalization; b) With Normalization.

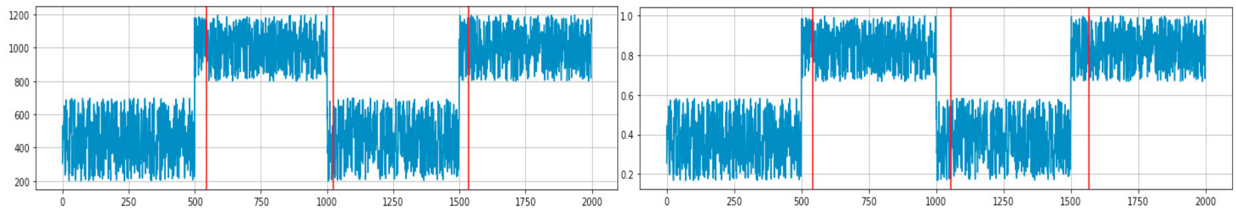

Fig. 7. ADWIN performance on recurrent drift a) Without Normalization; b) With Normalization.

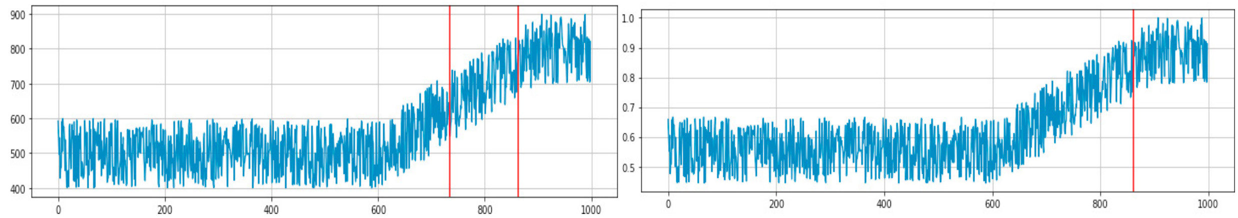

Fig. 8. ADWIN performance on gradual drift a) Without Normalization; b) With Normalization.

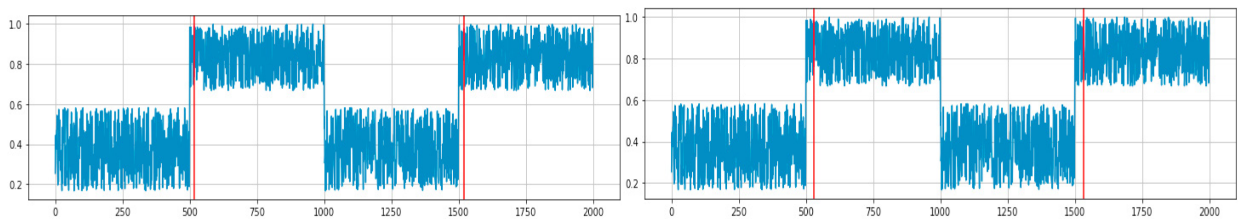

Fig. 9. HDDMs performance on recurrent drift with normalization a) HDDM_A performance; b) HDDM_W performance.

Table 9-10 shows the experimental results of time windowing, statistical control, and sequential analysis approaches with and without normalization. The performance of all approaches is evaluated using measures like true detection, total number of drifts detected and false positives. ADWIN performs well without normalization for all drift types. HDDMs are performing better with normalization; though it has failed to capture drifts that have down trends. Without normalization, the false positive rates are very high for all three scenarios for PH and HDDMs. DDM is capturing only with normalized data that too with false positive detections and failed to capture down trends. EDDM is not at all capturing any drifts in any scenarios.

Table 9: Total number of drifts detected by each algorithm without normalization

Total no of drifts detected (false positives in bracket with € 100)			
Drift detection Approaches	Without Normalization		
	Sudden drift	Recurring drift	Gradual drift
Actual drifts	2	3	2
ADWIN	2	3	2
PH	(49)	(68)	(33)
HDDM_A	(406)	(520)	(251)
HDDM_W	(589)	(536)	(278)

Table 10: Total number of drifts detected by each algorithm with normalization

Total no of drifts detected (false positives in bracket with € 100)			
Drift detection Approaches	With Normalization		
	Sudden drift	Recurring drift	Gradual drift
Actual drifts	2	3	2
ADWIN	2(1)	3	1
PH	(1)	(2)	(1)
HDDM_A	1	2	1
HDDM_W	0	2	0
DDM	(1)	2(1)	1

6. Conclusion

Machine learning models have been adopted by numerous organizations for decision making, hence reliable predictions are crucial for business success. Owing to the changing economic conditions, purchase behaviors and pandemic situations, models based on historical data become obsolete in anticipating future behaviors. The key difficulty is to detect the changes in real time, in a dynamic environment and making the model more accurate and restore its performance. In this work we introduced an integrated process of monitoring drift detection in real-time to react to changes quickly and to ensure models are performing to their full potential. Different distributions of data were analyzed for drift detection and the results show that the performance of ADWIN is better when compared to all other approaches in terms of number of drifts detected, less configuration and low false positive rate. It has been observed that normalization has significantly reduced the false positive rates of model outputs from PH and HDDMs which signifies the importance of efficient preprocessing steps for improved performance. Customized thresholds can be defined based on business knowledge where statistical approaches can also be configured to segment into warning\drifts. Future work is focused on evaluating elastic measures of similarity for detecting variations in data distributions.

References

[1] Baier, L., Kühl, N., Schöffer, J., & Satzger, G. (2020). Utilizing concept drift for measuring the effectiveness of policy interventions: The case of the COVID-19 pandemic. arXiv preprint arXiv:2012.03728.

[2] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346-2363.

[3] Chakravart, L., Laha, R. G., & Roy, J. W. (1967). Handbook of methods of applied statistics. Vol. I: techniques of computation, descriptive methods and statistical inference.

[4] Ross GJ, Adams NM, Tasoulis DK, Hand DJ (2012) Exponentially weighted moving average charts for detecting concept drift. Pattern Recognition Letters 33(2):191–198

[5] Lin, A. Z. (2017). Examining Distributional Shifts by Using Population Stability Index (PSI) for Model Validation and Diagnosis.

[6] Becker, A., & Becker, J. (2021). Dataset shift assessment measures in monitoring predictive models. *Procedia Computer Science*, *192*, 3391-3402.

[7] Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79-86.

[8] Hanif, M., & Seghouane, A. K. (2014, October). An effective image restoration using Kullback-Leibler divergence minimization. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 4522-4526). IEEE.

[9] Ponti, M., Kittler, J., Riva, M., de Campos, T., & Zor, C. (2017). A decision cognizant Kullback–Leibler divergence. *Pattern Recognition*, *61*, 470-478.

[10] Wang, L., Chen, S., & He, Q. (2021). Concept Drift-based Runtime Reliability Anomaly Detection for Edge Services Adaptation. *IEEE Transactions on Knowledge and Data Engineering*.

[11] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214-223). PMLR.

[12] (2014). Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 810-823.

[13] Baier, L., Schlör, T., Schöffer, J., & Kühl, N. (2021). Detecting concept drift with neural network model uncertainty. *arXiv preprint arXiv:2107.01873*.

[14] Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining* (pp. 443-448). Society for Industrial and Applied Mathematics.

[15] Bayram, F., Ahmed, B. S., & Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*, 108632.

[16] Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the drift-diffusion model in Python. Frontiers in neuroinformatics, 14.

[17] Pesaranghader, A., & Viktor, H. L. (2016, September). Fast hoeffding drift detection method for evolving data streams. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 96-111). Springer, Cham.

[18] Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., & Morales-Bueno, R. (2006, September). Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams* (Vol. 6, pp. 77-86).

[19] Ditzler, G., & Polikar, R. (2011, April). Hellinger distance-based drift detection for nonstationary environments. In *2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE)* (pp. 41-48). IEEE.

[20] Agrahari, S., & Singh, A. K. (2021). Concept Drift Detection in Data Stream Mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*.

[21] Barros, R. S., Cabral, D. R., Gonçalves Jr, P. M., & Santos, S. G. (2017). RDDM: Reactive drift detection method. *Expert Systems with Applications*, *90*, 344-355.

[22] Mouss, H., Mouss, D., Mouss, N., & Sefouhi, L. (2004, July). Test of page-hinckley, an approach for fault detection in an agro-alimentary production system. In *2004 5th Asian control conference (IEEE Cat. No. 04EX904)* (Vol. 2, pp. 815-818). IEEE.

[23] Regier, P., Briceño, H., & Boyer, J. N. (2019). Analyzing and comparing complex environmental time series using a cumulative sums approach. *MethodsX*, *6*, 779-787.

[24] Yi, F., & Qiu, P. (2022). An adaptive CUSUM chart for drift detection. *Quality and Reliability Engineering International*, *38*(2), 887-894.

[25] Grulich, P. M., Saitenmacher, R., Traub, J., Breß, S., Rabl, T., & Markl, V. (2018). Scalable Detection of Concept Drifts on Data Streams with Parallel Adaptive Windowing. In *EDBT* (pp. 477-480).

[26] Micevska, S. A Statistical Drift Detection Method.

[27] Flynn, T., & Yoo, S. (2019, December). Change detection with the kernel cumulative sum algorithm. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (pp. 6092-6099). IEEE.

[28] Duong, Q. H., Ramampiaro, H., & Nørvåg, K. (2018). Applying temporal dependence to detect changes in streaming data. *Applied Intelligence*, *48*(12), 4805-4823.

[29] Zhao, D., Bu, L., Alippi, C., & Wei, Q. (2017). A Kolmogorov-Smirnov test to detect changes in stationarity in big data. *IFAC-PapersOnLine*, *50*(1), 14260-14265.

[30] Pesaranghader, A., Viktor, H. L., & Paquet, E. (2018, July). McDiarmid drift detection methods for evolving data streams. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-9). IEEE.

[31] Brzeziński, D. (2010). Mining data streams with concept drift. *Cs Put Pozna*, *89*.

[32] Ashok, S., & Aravind, K. (2021, February). Impact of Covid-19 on Demand Planning: Building Resilient Forecasting Models. In 2021 The 5th International Conference on Compute and Data Analysis (pp. 59-66).