Model Drift Detection

Tianyi Sun
University of Chicago
tianyisun@uchicago.edu

ABSTRACT

As the advent of COVID-19 pandemic, AI is rapidly adapted by consumers and businesses. However, the reliability of machine learning models has hindered the success of AI systems. Machine learning model performance can fluctuate over time due to changes in the data input into the model after deployment. Which means that the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This is called concept drift (a.k.a model drift). Concept drift causes problems because the predictions become less accurate as time passes. Consequently, detecting concept drift early on is essential for maintaining up-to-date models in production that continuously provide value to company. Ideally, this should be incorporated as part of a robust framework for monitoring Machine Learning models in production.

In this work, we evaluate the effectiveness of six concept drift detection methods including ADaptive WINdowing (ADWIN) [2], Drift Detection Method (DDM) [1], Early Drift Detection Method (EDDM) [9], Drift Detection Method based on Hoeffding's bounds with moving average-test (HDDMA) [12], Drift Detection Method based on Hoeffding's bounds with moving weighted average-test (HDDMW) [3], Kolmogorov - Smirnov Windowing (KSWIN) [14], and Page-Hinkley (PageHinkley) [10] methods on the prediction result of eleven models including CatBoost (CB) [13], LightGBM (GBM), LightGBMLarge (GBML), LightGBMXT (GBMXT), KNeighborsDist (KNND), KNeighborsUnif (KNNU), Long Short-Term Memory (LSTM) [8], neural networks based FastAI (NNFAI), Random Forest (RF) [4], XGBoost (XGB) [6], and Extra Trees (XT) [7]. We also evaluate to see how does signal processing techniques including frequency decomposition and low pass filtering affect the result of concept drift detection.

KEYWORDS

Concept Drift, Model Drift, Machine Learning

1 INTRODUCTION

Drift is the change in an entity with respect to a baseline. Data Drift is defined as a change in the distribution of data, which underlies model drift. In the case of production Machine Learning models, this is the change between the real-time production data and a baseline data set, i.e. the training set, that is representative of the task the model is intended to preform. Production data can diverge or drift from the baseline data over time due to changes in the real world. Drift of the predicted values is a good proxy for concept drift, or data integrity issues and can inform model re-train cadence.

There are several types of Machine Learning drift:

- Concept drift (model drift) is defined as a change in the distribution $P(y|X)$, where y is the real label, and X are the available features. It is a shift in the actual relationship between the model's inputs and the output. It is a change in $P(Y|X)$.
- Prediction drift is a shift in the model's predictions. It is a change in $P(\hat{Y}|X)$.
- Label drift is a shift in the model's output or label distribution. It is a change in $P(Y)$.
- Feature drift is a shift in the model's input data distribution. It is a change in $P(X)$.

Concept drift is a discrepancy between a real and learned decision boundary. It is necessary to re-learn the data to maintain the accuracy of the previous regime. If ground truth labels are available and sufficiently real-time, performance drift is the strongest indicator of this. In the absence of real-time ground truth, drift in prediction and feature distributions are often indicative of important changes in the world. Unlike performance-drift, however, it is possible for these quantities to drift with respect to an accurately modeled decision boundary. In that case, model performance will be unchanged.

The problem with concept drift is that the core assumption of Machine Learning is that the training distribution reflects the "real-world" distribution, otherwise nothing ensures that the trained model is fit for the target task. Concept drift is a central reason for the need to refresh and retrain Machine Learning models. As the incoming data drifts away from the historical data which was used for training, the relationships and correlations between features changes as well.

A popular choice of concept drift detection algorithm for streaming data is the ADWIN. Some popular choices of concept drift detection algorithm for batched data are the KSWIN, the chi-squared test, and the adversarial validation.

1.1 Contribution:

- We find out that KSWIN is the best model drift detection method in our case.
- We find out that the low pass filtering technique is helpful to accurately detect model drift in our case.

2 METHOD

In this section, we introduce how concept drift detection methods work.

2.1 ADWIN

ADaptive WINdowing is a popular drift detection method with mathematical guarantees. ADWIN keeps a variable - length window of recent items. It holds that there has no been change in the data distribution. This window is further divided into two sub-windows (W_0, W_1) used to determine if a change has happened. ADWIN compares the average of W_0 and W_1 to confirm that they correspond

Project Report, Pytorch,
2021.

to the same distribution. Model drift is detected if the distribution equality no longer holds. Upon detecting a drift, W_0 is replaced by W_1 and a new W_1 is initialized. ADWIN uses a confidence value $\eta =\in (0, 1)$ to determine if the two sub-windows correspond to the same distribution.

2.2 DDM

Drift Detection Method is a concept change detection method based on the PAC learning model premise, that the learner's error rate will decrease as the number of analysed samples increase, as long as the data distribution is stationary. If the algorithm detects an increase in the error rate, which surpasses a threshold, $(p_i + s_i)$, where p_i is the error rate at instant i and s_i is the standard deviation at instant i, then either change is detected or the algorithm will warn the user that change may occur in the near future, which is called the warning zone. The condition for entering the warning zone is

$$p_i + s_i \geq p_{min} + 2 \times s_{min},$$

and detecting change is

$$p_i + s_i \geq p_{min} + 3 \times s_{min}.$$

The minimum recorded error rate is represented as p_{min} and the minimum recorded standard deviation is represented as s_{min}.

2.3 EDDM

Early Drift Detection Method aims to improve the detection rate of gradual concept drift in DDM, while keeping a good performance against abrupt concept drift. It keeps track of the average distance between two errors instead of only the error rate. For this, it is necessary to keep track of the running average distance and the running standard deviation, as well as the maximum distance and the maximum standard deviation.

The algorithm is similar to the DDM algorithm. It works with the running average distance and the running standard deviation, as well as the values of $p_i^{'}$ and $s_i^{'}$ when $(p_i' + 2 \times s_i')$ reaches its maximum.

Like DDM, there are two threshold values that define the borderline among no change, warning zone, and drift detected. For example,

$$\frac{p_i' + 2 \times s_i'}{p_{max}' + 2 \times s_{max}'} < \alpha$$

and

$$\frac{p_i' + 2 \times s_i'}{p_{max}' + 2 \times s_{max}'} < \beta,$$

where $\alpha = 0.95$ and $\beta = 0.9$

2.4 HDDMA

Drift Detection Method based on Hoeffding's bounds with moving average-test is a method based on the Hoeffding's inequality. HDDMA uses the average as estimator. It receives as input a stream of real values and returns the estimated status of the stream: STABLE, WARNING or DRIFT.

2.5 HDDMW

Drift Detection Method based on Hoeffding's bounds with moving weighted average-test is a method based on McDiarmid's bounds. HDDMW uses the Exponentially Weighted Moving Average (EWMA) statistic as estimator. It receives as input a stream of real predictions and returns the estimated status of the stream: STABLE, WARNING or DRIFT.

2.6 KSWIN

Kolmogorov - Smirnov Windowing is a method based on the Kolmogorov - Smirnov statistical test. KS-test is a statistical test with no assumption of underlying data distribution. KSWIN can monitor data or performance distributions. Note that the detector accepts one dimensional input as array.

Kolmogorov - Smirnov Windowing maintains a sliding Ψ of fixed window size n. The last state size r samples of Ψ are assumed to represent the last concept considered as R. From the first $n - r$ samples of Ψ, r samples are uniformly drawn, representing an approximated last concept W.

The KS-test is performed on the window R and W of the same size. KS-test compares the distance of the empirical cumulative data distribution $dist(R, W)$.

A concept drift is detected by KSWIN, if

$$dist(R, W) > \sqrt{-\frac{ln\alpha}{r}}.$$

The difference in empirical data distribution between the windows R and W is too large since R and W come from the same distribution.

2.7 PageHinkley

Page-Hinkley method works by computing the observed values and their mean up to the current moment. Page-Hinkley does not signal warning zones, only change detection. The method works by means of the Page-Hinkley test. In general lines it will detect a concept drift if the observed mean at some instant is greater than a threshold value lambda.

3 DATA

The data we use to evaluate the effectiveness of the model drift detection methods is the outputs of models, include GBM, CB, GBML, GBMXT, KNND, KNNU, LSTM, NNFAI, RF, XGB, and XT in order, trained on important features. So we have eleven output sequences in total. The sequence of GBM model output is shown in Figure 2. The other model outputs can be found in APPENDIX.

4 EXPERIMENTAL EVALUATION

In this section, we evaluate the effectiveness of the purposed six drift detection methods and the usefulness of signal processing techniques on model drift detection. The purpose of drift detection method is to identify the data distribution changes. A good drift detection method is the one that maximize the true positives while keeping the number of false positives to a minimum.

We use the purposed six drift detection methods, including ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley, to detect drifts on the GBM model output. Figure 1 shows the results of the six drift detection methods on original data. We then use low

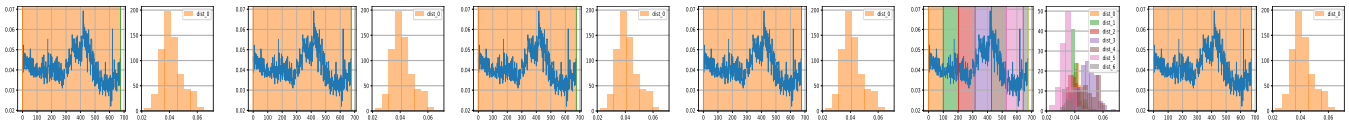

Figure 1: GBM model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

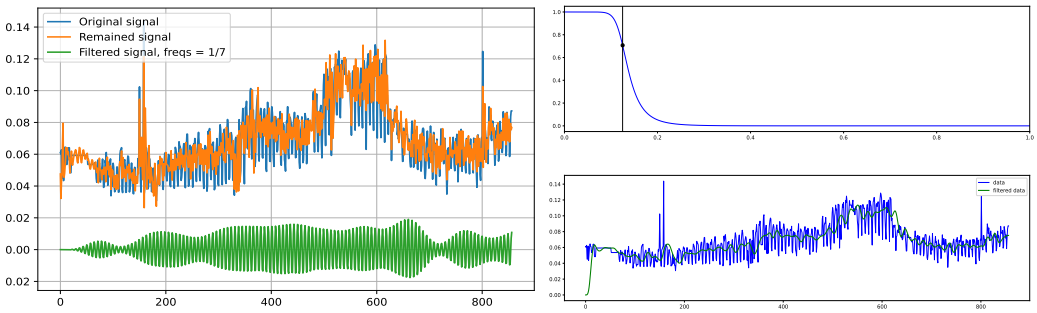

Figure 2: The figure on the left is the low pass filtered remaining GBM model output and the figure on the right is the frequency decomposed GBM model output.

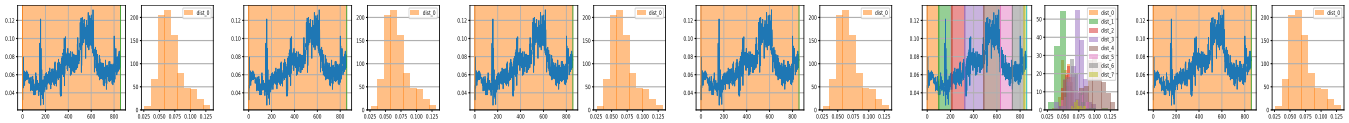

Figure 3: Drift detection on the low pass filtered remaining GBM model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

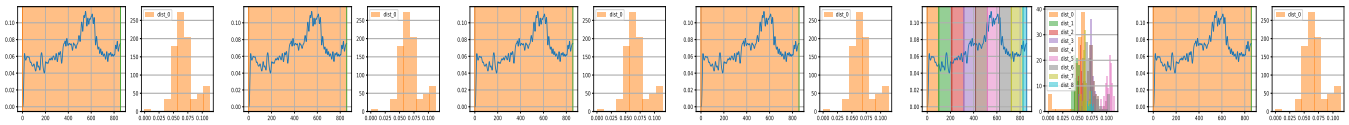

Figure 4: Drift detection on the frequency decomposed GBM model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

pass filtering and frequency decomposition techniques to process the original GBM model output. Figure 2 shows how the processed GBM model output looks like. Then we used the six model drift detection methods to detect on the processed GBM model output. The detection results are shown in Figure 3 and Figure 4 for the low pass filtered remaining GBM model output and the frequency decomposed GBM model output, respectively.

Then we used the same procedure illustrated above to detect model drifts on outputs of CB, GBML, GBMXT, KNND, KNNU, LSTM, NNFAI, RF, XGB, and XT models. APPENDIX shows the effectiveness of the six model drift detection methods on those model outputs and the usefulness of signal processing techniques on model drift detection.

5 CONCLUSION

By comparing the drift detection results in Figure 1, Figure 3, Figure 4, and APPENDIX, we find that KSWIN clearly detect out the distribution changes in the given outputs. By comparing the difference between Figure 1, which is the model drift detection result of original data, and Figure 3, which is the model drift detection result of low pass filtered remaining data, and the difference between Figure 1 and Figure 4, which is the model drift detection result of frequency decomposed data, we find out that the similarity between the KSWIN detection result on the original data and the low pass filtered remaining data is higher than the similar between the KSWIN detection result on the original data and the frequency decomposed data. This is also true in APPENDIX.

Therefore, KSWIN is the most effective model drift detection method and the low pass filtering techniques is useful for accurately detect the drift.

6 FUTURE WORK

There are several future works that we are planing to do:

- We will further investigate why KSWIN is the best fit for our data and the other methods are not. We will look into

the distribution of our data output and the inner structure of the six drift detection algorithms.
- We will use a topological tool [16] to detect concept drift. Specifically, using the Explainable Boosting Machine (EBM) from InterpretML [11] in combination with the Mapper [15] from Topological Data Analysis (TDA) [5].

REFERENCES

[1] Marcel Altendeitering and Stephan Dºbler. 2020. Scalable Detection of Concept Drift: A Learning Technique Based on Support Vector Machines. *Procedia Manufacturing* 51 (2020), 400–407. https://doi.org/10.1016/j.promfg.2020.10.057 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

[2] Albert Bifet and Ricard Gavaldà. 2007. Learning from Time-Changing Data with Adaptive Windowing. In *SDM*.

[3] Isvani Inocencio Frías Blanco, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales Bueno, Agustín Alejandro Ortiz Díaz, and Yailé Caballero Mota. 2015. Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds. *IEEE Transactions on Knowledge and Data Engineering* 27 (2015), 810–823.

[4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

[5] Gunnar E. Carlsson. 2009. Topology and data. *Bull. Amer. Math. Soc.* 46 (2009), 255–308.

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug 2016). https://doi.org/10.1145/2939672.2939785

[7] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely Randomized Trees. *Mach. Learn.* 63, 1 (apr 2006), 3–42. https://doi.org/10.1007/s10994-006-6226-1

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[9] Manuel Baena-Garca Jose, José Del Campo-Ávila, Raúl Fidalgo, Albert Bifet, Ricard Gavaldà, and Rafael Morales-bueno. [n. d.]. Early Drift Detection Method.

[10] Nisrine Jrad, Amar Kachenoura, Anca Nica, Isabelle Merlet, and Fabrice Wendling. 2017. A Page-Hinkley based method for HFOs detection in epileptic depth-EEG. In *2017 25th European Signal Processing Conference (EUSIPCO)*. 1295–1299. https://doi.org/10.23919/EUSIPCO.2017.8081417

[11] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. 2019. InterpretML: A Unified Framework for Machine Learning Interpretability. arXiv:cs.LG/1909.09223

[12] Ali Pesaranghader and Herna L. Viktor. 2016. Fast Hoeffding Drift Detection Method for Evolving Data Streams. In *ECML/PKDD*.

[13] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. CatBoost: unbiased boosting with categorical features. arXiv:cs.LG/1706.09516

[14] Christoph Raab, Moritz Heusinger, and Frank-Michael Schleif. 2020. Reactive Soft Prototype Computing for Concept Drift Streams. *Neurocomputing* 416 (Nov 2020), 340–351. https://doi.org/10.1016/j.neucom.2019.11.111

[15] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Eurographics Symposium on Point-Based Graphics*, M. Botsch, R. Pajarola, B. Chen, and M. Zwicker (Eds.). The Eurographics Association. https://doi.org/10.2312/SPBG/SPBG07/091-100

[16] Hendrik Jacob van Veen. 2020. Novel Topological Shapes of Model Interpretability. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*. https://openreview.net/forum?id=G-kWQ9WvBMq

1 APPENDIX

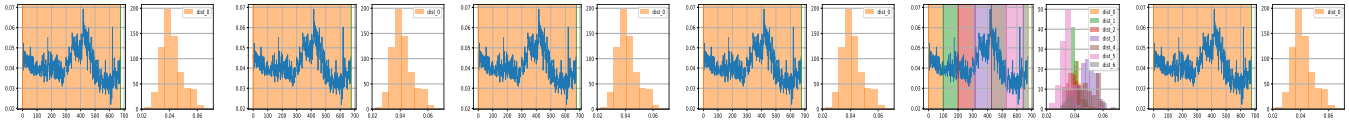

Figure 1: CB model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

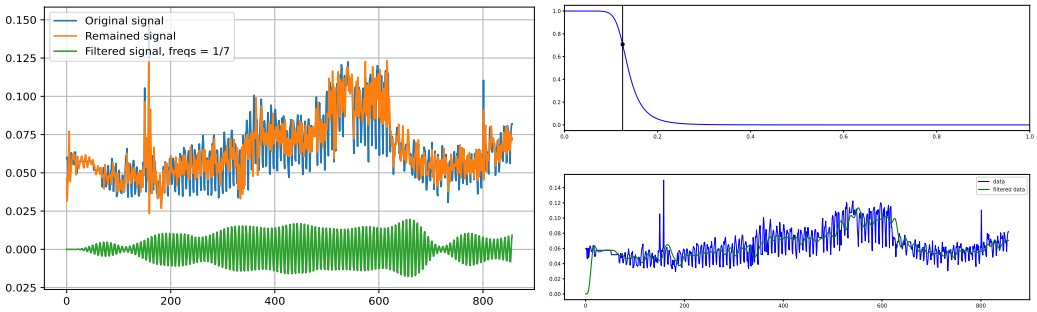

Figure 2: The figure on the left is the low pass filtered remaining CB model output and the figure on the right is the frequency decomposed CB model output.

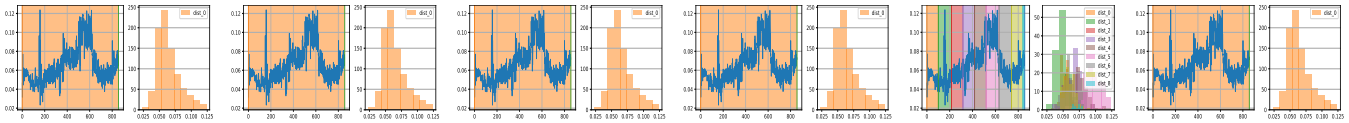

Figure 3: Drift detection on the low pass filtered remaining CB model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

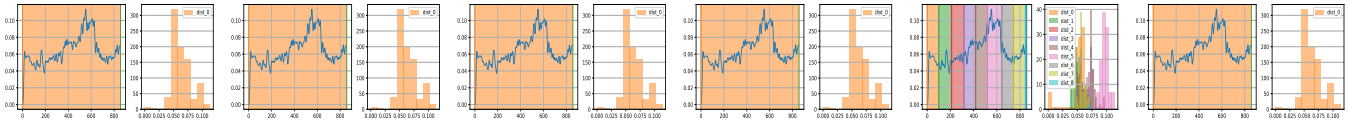

Figure 4: Drift detection on the frequency decomposed CB model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

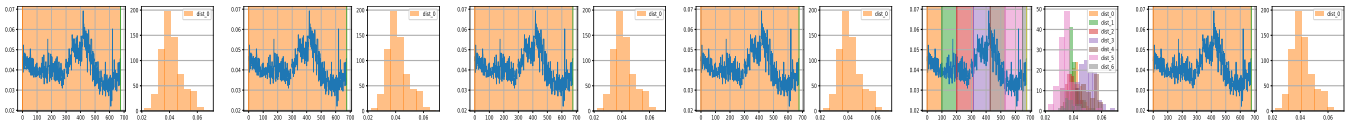

Figure 5: GBML model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

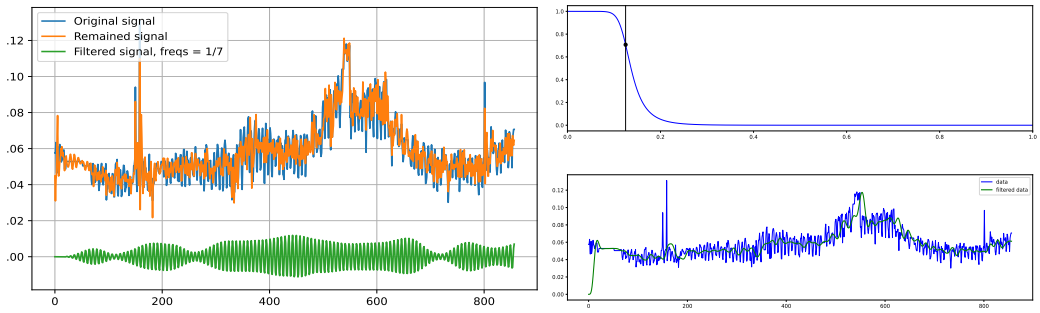

Figure 6: The figure on the left is the low pass filtered remaining GBML model output and the figure on the right is the frequency decomposed GBML model output.

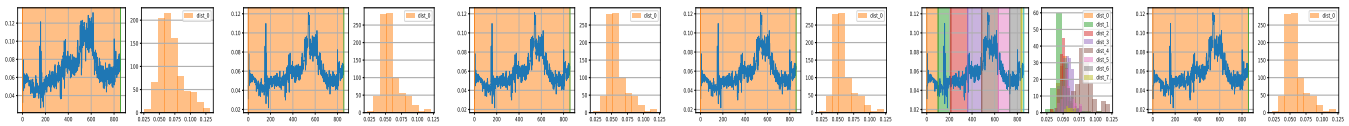

Figure 7: Drift detection on the low pass filtered remaining GBML model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

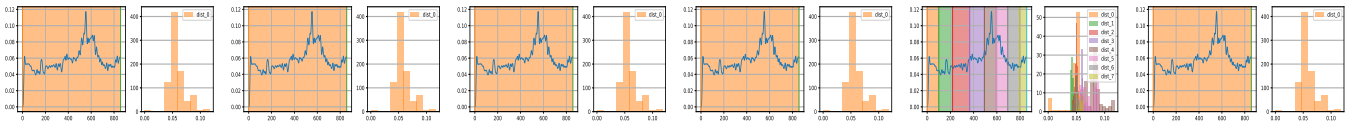

Figure 8: Drift detection on the frequency decomposed GBML model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

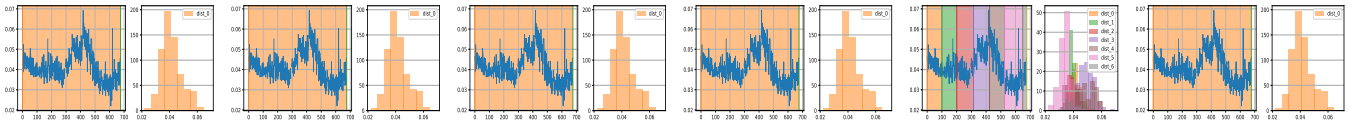

Figure 9: GBMXT model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

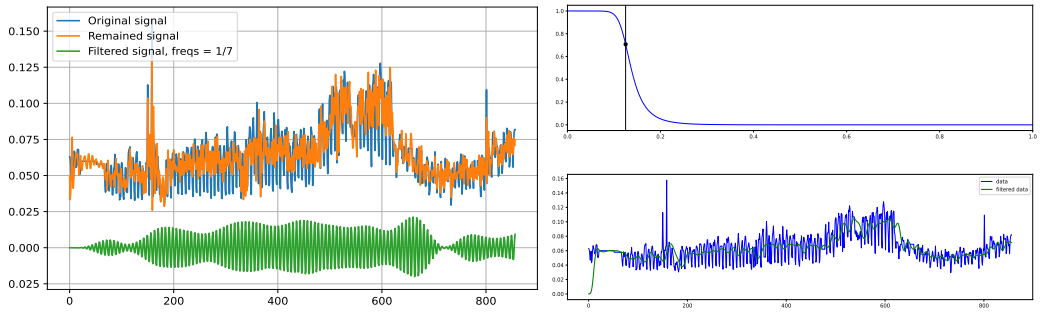

Figure 10: The figure on the left is the low pass filtered remaining GBMXT model output and the figure on the right is the frequency decomposed GBMXT model output.

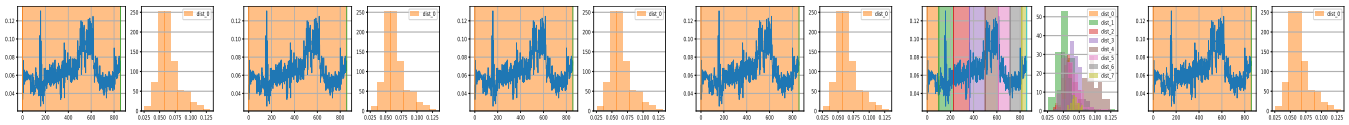

Figure 11: Drift detection on the low pass filtered remaining GBMXT model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

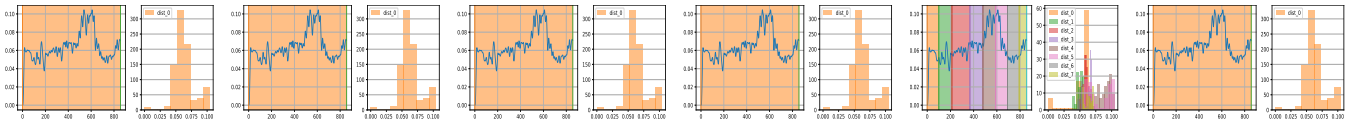

Figure 12: Drift detection on the frequency decomposed GBMXT model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

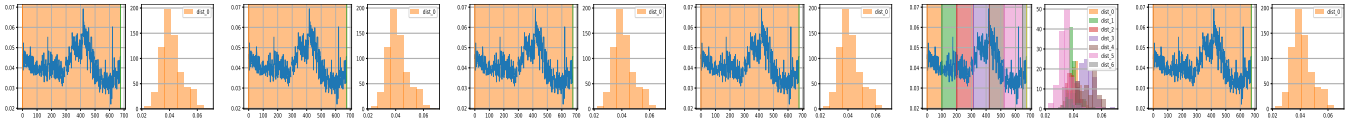

Figure 13: KNND model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

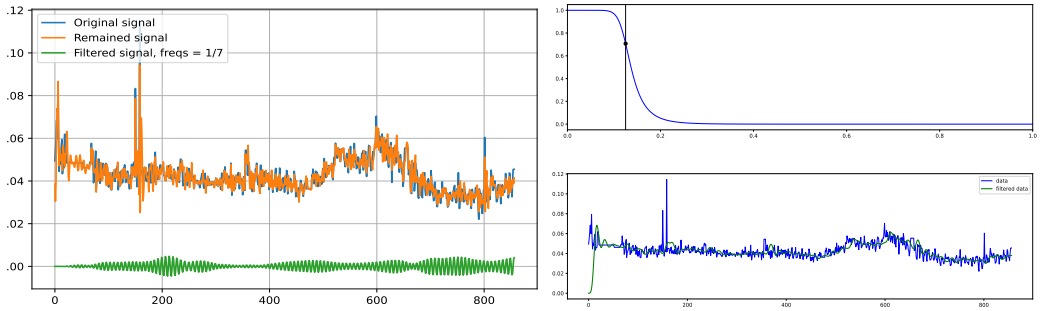

Figure 14: The figure on the left is the low pass filtered remaining KNND model output and the figure on the right is the frequency decomposed KNND model output.

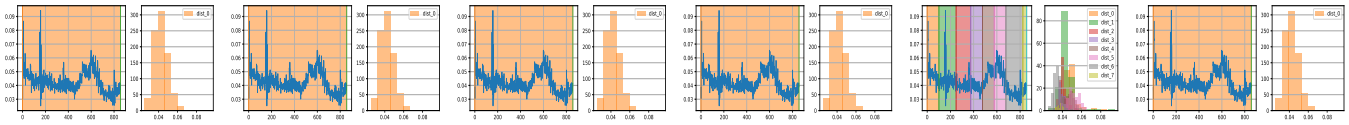

Figure 15: Drift detection on the low pass filtered remaining KNND model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

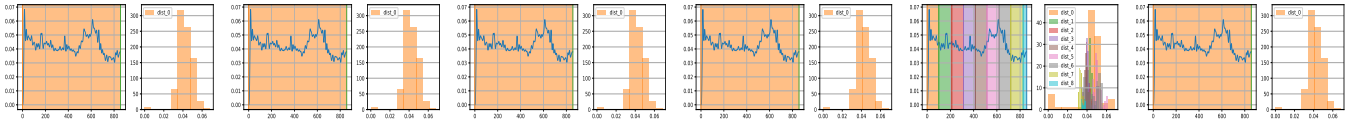

Figure 16: Drift detection on the frequency decomposed KNND model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

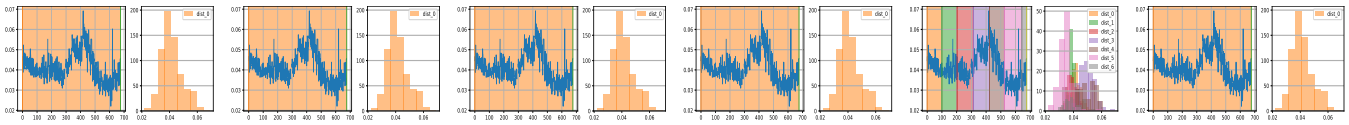

Figure 17: KNNU model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

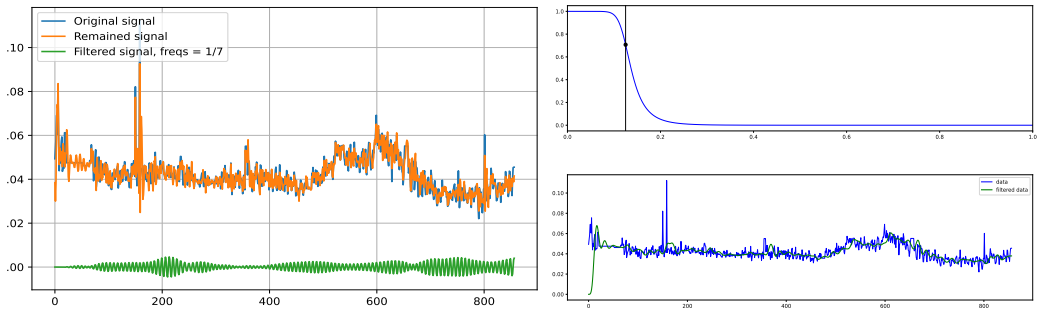

Figure 18: The figure on the left is the low pass filtered remaining KNNU model output and the figure on the right is the frequency decomposed KNNU model output.

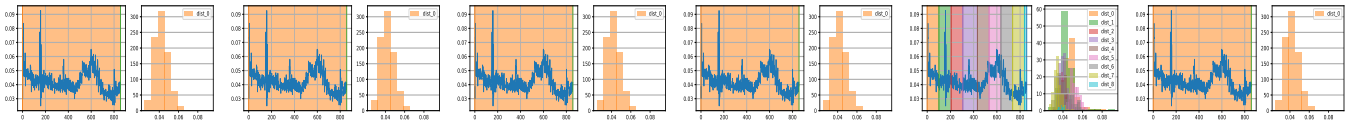

Figure 19: Drift detection on the low pass filtered remaining KNNU model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

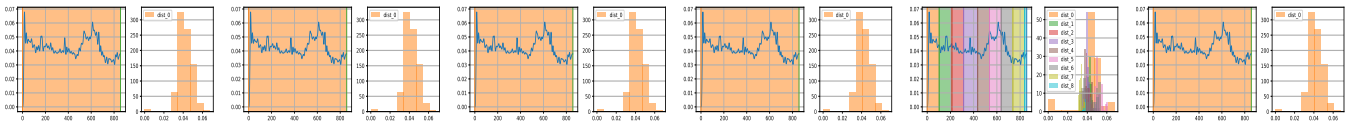

Figure 20: Drift detection on the frequency decomposed KNNU model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

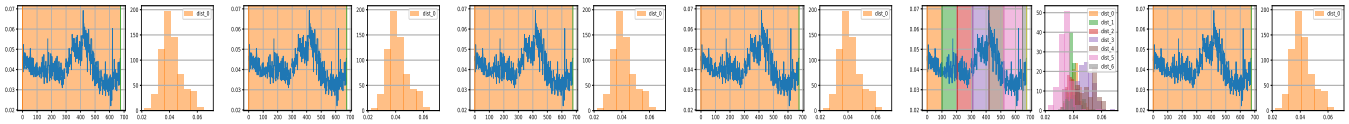

Figure 21: LSTM model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

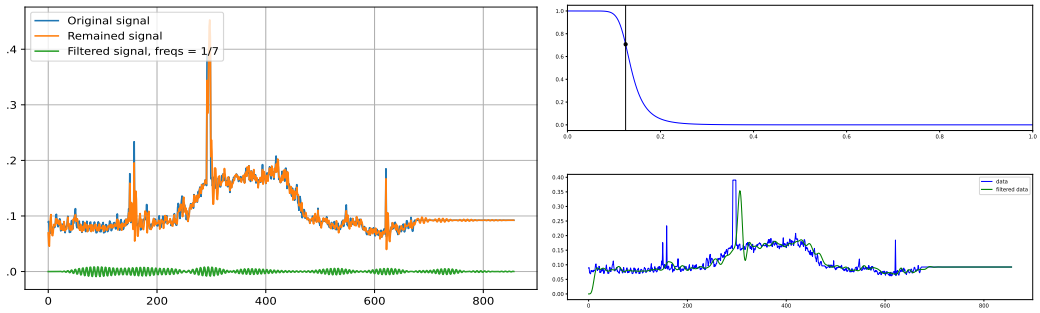

Figure 22: The figure on the left is the low pass filtered remaining LSTM model output and the figure on the right is the frequency decomposed LSTM model output.

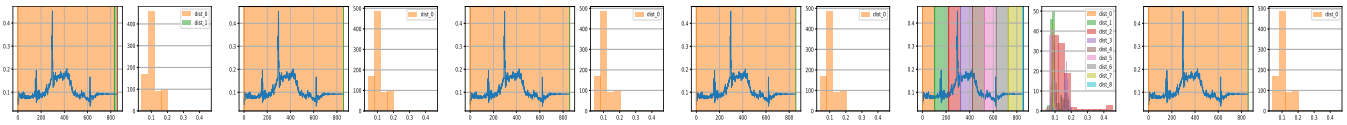

Figure 23: Drift detection on the low pass filtered remaining LSTM model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

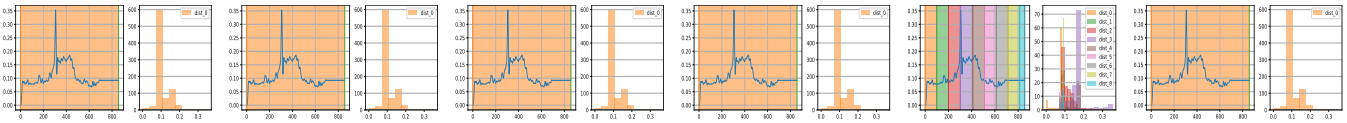

Figure 24: Drift detection on the frequency decomposed LSTM model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

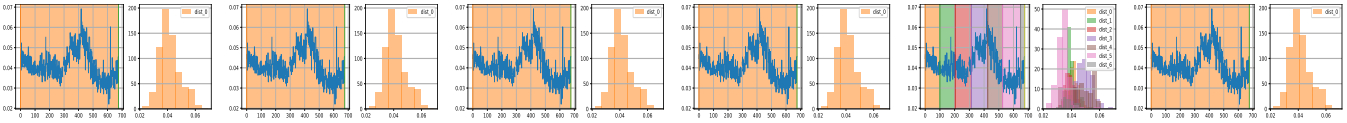

Figure 25: NNFAI model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

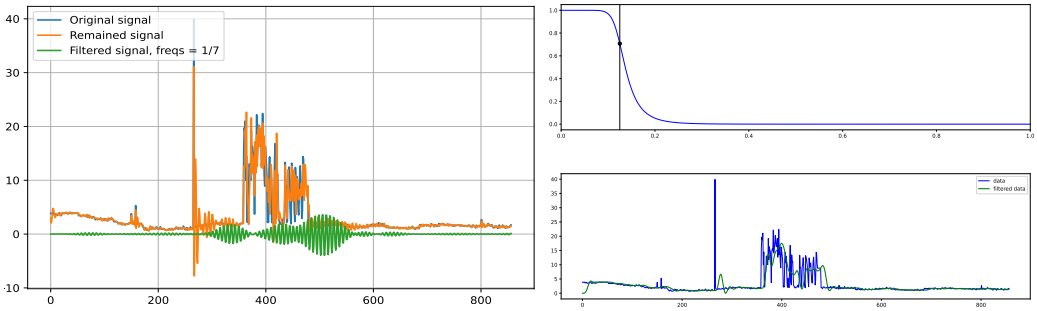

Figure 26: The figure on the left is the low pass filtered remaining NNFAI model output and the figure on the right is the frequency decomposed NNFAI model output.

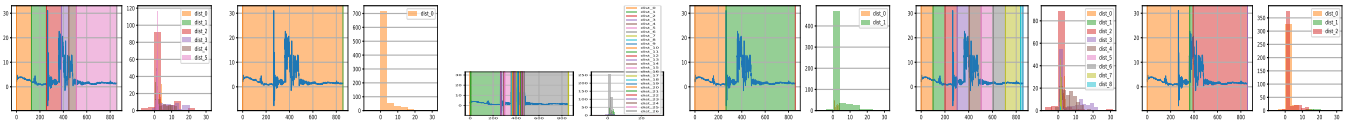

Figure 27: Drift detection on the low pass filtered remaining NNFAI model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

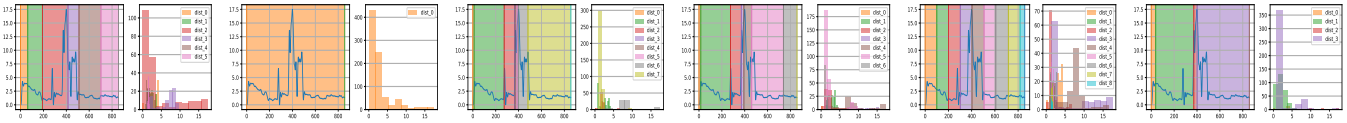

Figure 28: Drift detection on the frequency decomposed NNFAI model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

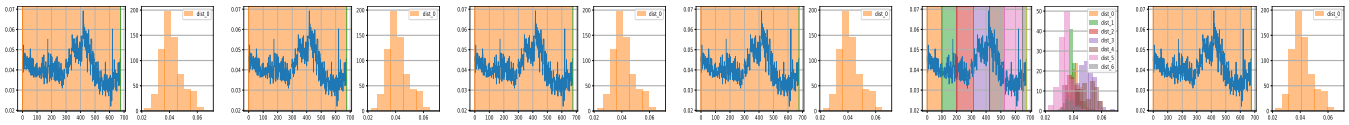

Figure 29: RF model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

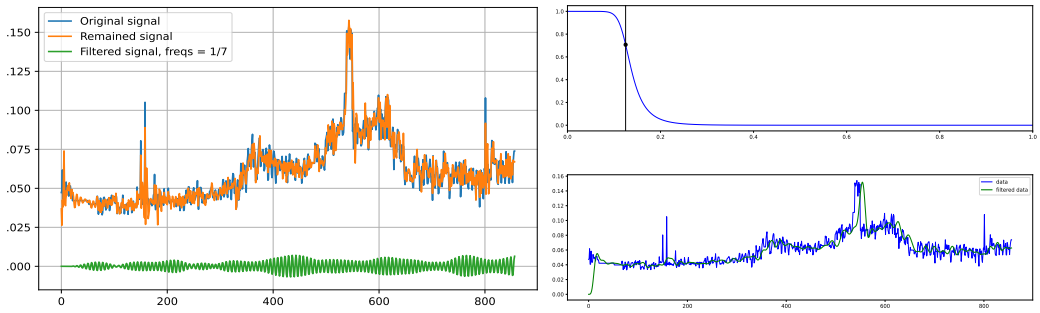

Figure 30: The figure on the left is the low pass filtered remaining RF model output and the figure on the right is the frequency decomposed RF model output.

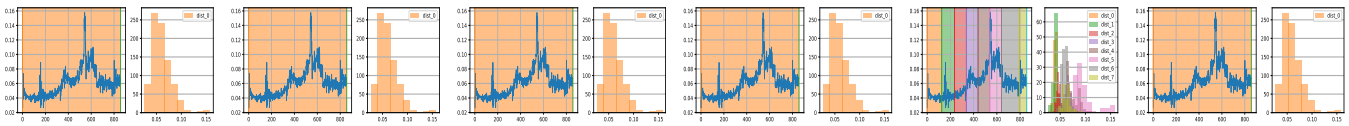

Figure 31: Drift detection on the low pass filtered remaining RF model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

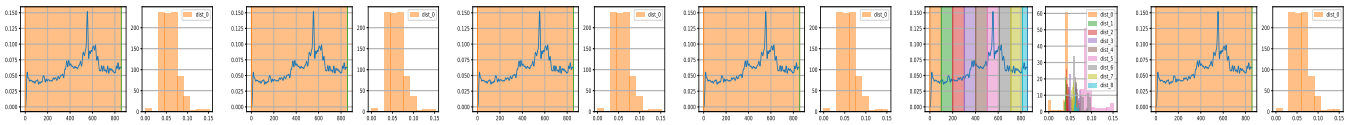

Figure 32: Drift detection on the frequency decomposed RF model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

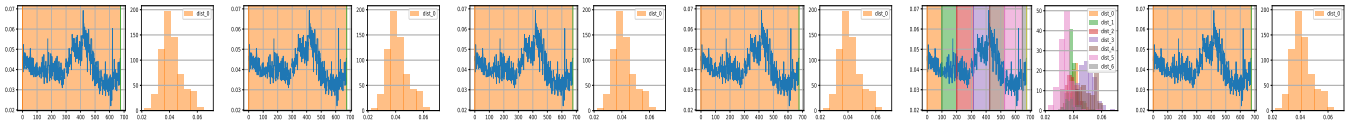

Figure 33: XGB model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

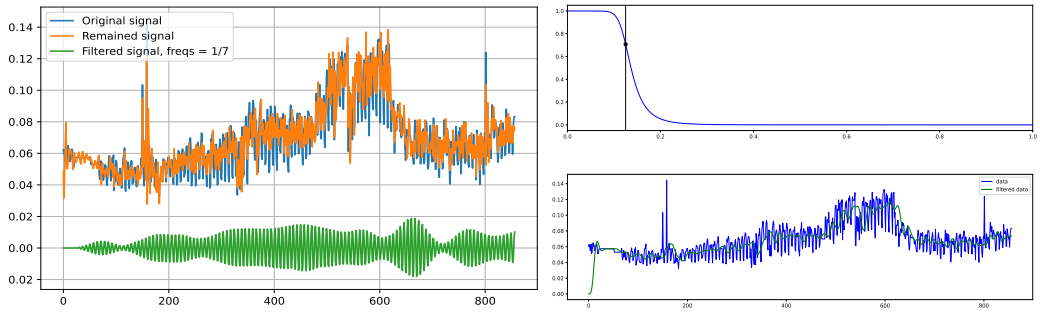

Figure 34: The figure on the left is the low pass filtered remaining XGB model output and the figure on the right is the frequency decomposed XGB model output.

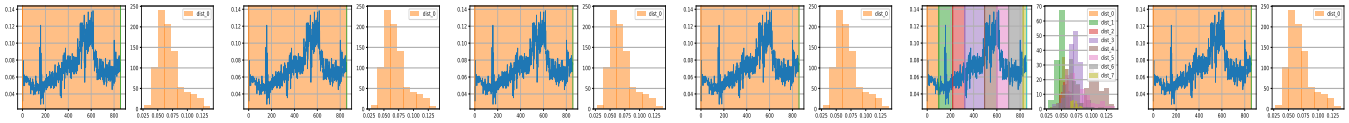

Figure 35: Drift detection on the low pass filtered remaining XGB model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

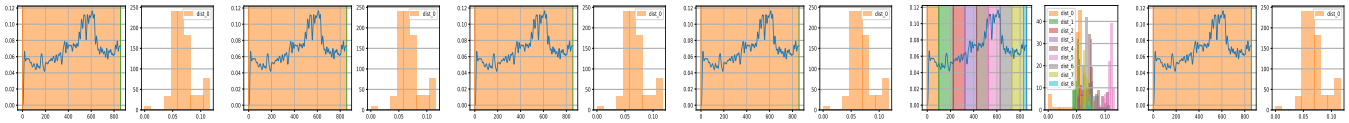

Figure 36: Drift detection on the frequency decomposed XGB model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

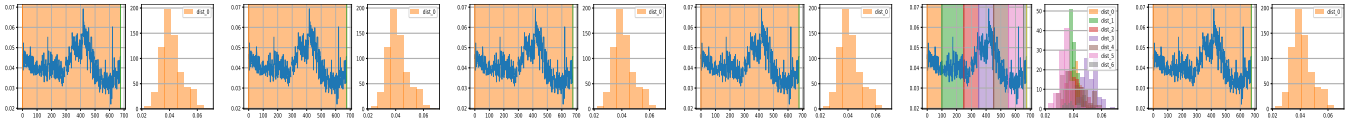

Figure 37: XT model drift detection using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

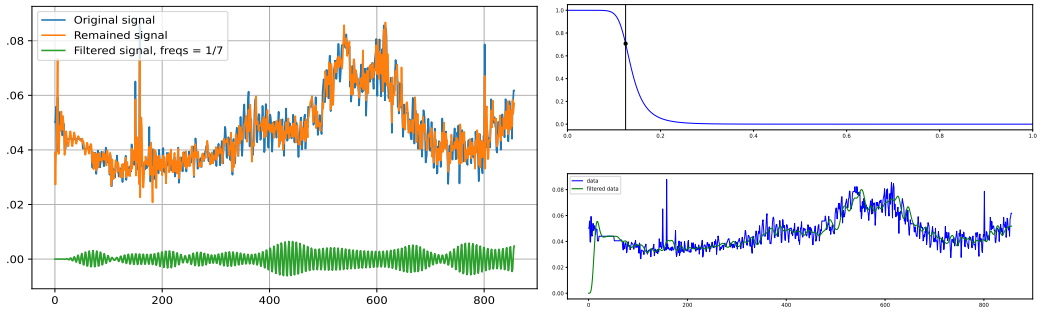

Figure 38: The figure on the left is the low pass filtered remaining XT model output and the figure on the right is the frequency decomposed XT model output.

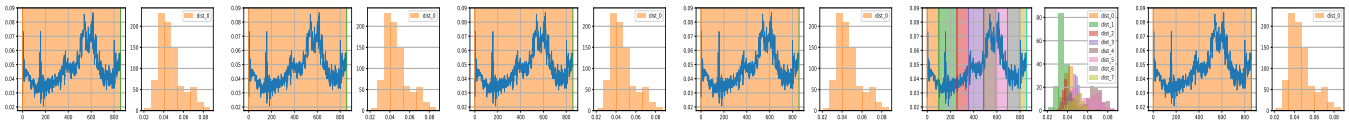

Figure 39: Drift detection on the low pass filtered remaining XT model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.

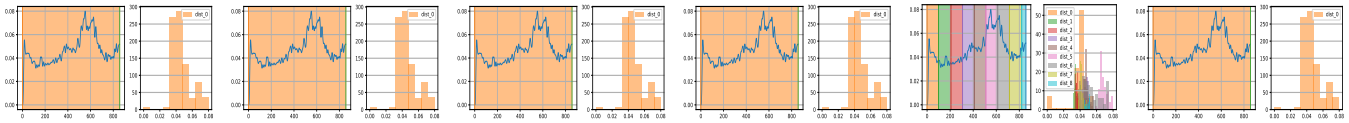

Figure 40: Drift detection on the frequency decomposed XT model output using ADWIN, EDDM, HDDMA, HDDMW, KSWIN, and PageHinkley methods in order.