Quantifying input data drift in medical machine learning models by detecting change-points in time-series data

Smriti Prathapan[a], Ravi K. Samala[a], Nathan Hadjiyski[a], Pierre-François D'Haese[b], Fabien Maldonado[c], Phuong Nguyen[d], Yelena Yesha[d,e], and Berkman Sahiner[a]

[a]U.S. Food and Drug Administration, Center for Devices and Radiological Health, Office of Science and Engineering Laboratories, Silver Spring, MD
[b]Rockefeller Neuroscience Institute, West Virginia University, Morgantown, WV
[c]Vanderbilt University Medical Center, Nashville, TN
[d]University of Miami, Department of Computer Science, Coral Gables, FL
[e]University of Miami, Department of Radiology, Coral Gables, FL

ABSTRACT

Devices enabled by artificial intelligence (AI) and machine learning (ML) are being introduced for clinical use at an accelerating pace. In a dynamic clinical environment, these devices may encounter conditions different from those they were developed for. The statistical data mismatch between training/initial testing and production is often referred to as *data drift*. Detecting and quantifying data drift is significant for ensuring that AI model performs as expected in clinical environments. A drift detector signals when a corrective action is needed if the performance changes. In this study, we investigate how a change in the performance of an AI model due to data drift can be detected and quantified using a cumulative sum (CUSUM) control chart. To study the properties of CUSUM, we first simulate different scenarios that change the performance of an AI model. We simulate a sudden change in the mean of the performance metric at a change-point (change day) in time. The task is to quickly detect the change while providing few false-alarms before the change-point, which may be caused by the statistical variation of the performance metric over time. Subsequently, we simulate data drift by denoising the Emory Breast Imaging Dataset (EMBED) after a pre-defined change-point. We detect the change-point by studying the pre- and post-change specificity of a mammographic CAD algorithm. Our results indicate that with the appropriate choice of parameters, CUSUM is able to quickly detect relatively small drifts with a small number of false-positive alarms.

Keywords: Medical Imaging, Mammography, Drift Detection, CUSUM, Clinical AI workflow, Average Run Length, Quality assurance

1. INTRODUCTION

The term *data drift*, as noted by Moreno-Torres *et al.*,[1] refers to a difference between the training and the test data distributions, where data includes both the input to the model and the target variable. In the medical machine learning literature, this translates to a difference in the data distribution that was used for AI model development and the clinical use. Data drift is categorized based on the changes in : (i) the distribution of the model input X, (ii) the target variable Y and (iii) the conditional distribution of $Y|X$ or $X|Y$.[2]

In the context of AI-enabled computer-aided diagnosis (CAD) system evaluation, we are interested in data drift because it may result in a change in the performance of the CAD system. The focus of this work is to understand the trade-offs among how quickly a change in the AI model performance can be detected, the magnitude of the change with respect to inherent variability, and how often a false-alarm is produced for a performance change. We use cumulative sum (CUSUM) control charts to detect a change in CAD performance in time. We use numerical simulation studies, as well as a change induced by filtering mammograms, to study the trade-offs.

Further author information: (Send correspondence to Berkman Sahiner)
Berkman Sahiner: E-mail: Berkman.Sahiner@fda.hhs.gov

Medical Imaging 2024: Computer-Aided Diagnosis, edited by Weijie Chen,
Susan M. Astley, Proc. of SPIE Vol. 12927, 129270E · © 2024
SPIE · 1605-7422 · doi: 10.1117/12.3008771

2. METHODS

2.1 Cumulative Sum (CUSUM) Control Charts

The goal in monitoring clinical AI models is to detect a performance change and raise an alarm to take corrective action. To monitor a one-dimensional performance measure x, we demonstrate the use of the Cumulative Sum (CUSUM) control chart that provides a numerical output to identify a performance change.

CUSUM control chart was originally developed by Page[3] for industrial process control. Borrowing terms from statistical process control, a process is said to be in-control when it produces output within a certain range of variation that is considered acceptable and consistent with its designed specifications, or when the observed variation reflects random fluctuations.[4] The process is said to be out-of-control otherwise. The mean when a process is in-control is called in-control mean, (μ_0 = target mean) and the mean when the process is out-of-control is called the out-of-control mean (μ_1). The CUSUM method detects drifts in the mean of a process by aggregating deviations from the in-control mean. A process is declared out-of-control when drift in mean persists. CUSUM-based analysis is considered to be one of the optimal statistical tools for quality control in the health care domain due to its ability to detect small changes quickly.[5] While there are different types of CUSUM charts, in this study we focus on two-sided CUSUM, highlighting whether positive or negative cumulative values are more relevant for detecting the change-point at a given time.

A two-sided CUSUM control chart is created by computing the deviation from the in-control mean as in Equation (1):

$$\begin{aligned} S_{hi}(i) &= max(0, S_{hi}(i-1) + x_i - \hat{\mu_0} - K) \\ S_{lo}(i) &= max(0, S_{lo}(i-1) + \hat{\mu_0} - K - x_i) \end{aligned} \tag{1}$$

where i denotes a unit of time, x_i is the value of quantity being monitored at time i, $\hat{\mu_0}$ is the in-control mean of x_i, and K is a variable called "reference value" that is related to the magnitude of change that one is interested in detecting. S_{hi} and S_{lo} are the cumulative sum of positive and negative changes. A CUSUM scheme thus cumulates deviations more than K units from the in-control mean value. Let σ denote the in-control standard deviation of x_i. If one is interested in detecting a change in the the mean of x_i that is equal to σ, then a typical choice for K is $K = 0.5\sigma$. More generally, the normalized reference value k is defined with the relation $K = k\sigma$, with $k = 0.5$ as the default choice for detecting a unit standard deviation change in the mean of the process.

In CUSUM, the process is considered to be in-control as long as both S_{hi} and S_{lo} are less than a decision limit (also referred to as threshold) *H*. Similar to K, H is typically expressed in terms of the standard deviation of x_i, $H = h\sigma$, where h is referred to as the normalized threshold. In typical applications, h may default to 4,[6,7] and an alarm is produced when either S_{hi} or S_{lo} exceeds more than 4 times the standard deviation of the process.

2.2 Drift detection using CUSUM

We describe a method to detect a change in the performance of a machine learning (ML) model (measured by metrics such as area under the receiver operating characteristic (ROC) curve (AUC) and specificity) using a two-sided CUSUM chart. Let *d=0,...n-1* indicate the days (or time) during which the ML model performance is observed. We assume that the performance is in-control between days 0 and d_s, $d_s < n$, and a change in performance occurs on day d_s such that $x_{0...d_{s-1}}$ and $x_{d_s...d_{n-1}}$ are performance measure values for the pre-change and the post-change periods respectively.

In this setting, three different cases are of interest:

(i) Change-point detected at $d_d : d_d \geq d_s$ with a detection delay $d_d - d_s$.
(ii) Change-point detected at $d_d : d_d < d_s$. This is called a false alarm (Type I error).
(iii) Change-point is not detected. This is a missed detection (Type II error).

2.2.1 Performance Assessment

The sensitivity of CUSUM to detect performance drift is evaluated based on the occurrence of type-I errors measured using the Mean Time Between False Alarms (MTBFA) and Average Detection Delay (ADD). In our study, MTBFA and ADD are averaged from the outcome of a series of N independent experiments where the change-point detection is performed until the conclusion of the entire simulation time (days), n.

Mean time between false alarms (MTBFA), as defined by Page[8] and Lorden,[9] evaluates the average number of days before a false alarm is detected. MTBFA is given as the expected value of d_d in the pre-change regime as $MTBFA = \mathbb{E}_0(d_d)$, where the subscript 0 indicates the pre-change regime. The estimate of MTBFA is complicated by the fact that in many experiments, no false positives are observed in the in-control regime, i.e., the data is right-censored at the start of change-point, d_s. We therefore use the maximum-likelihood estimate for MTBFA under right-censoring,[10] given as the ratio of the total exposure to risk and the number of detections as:

$$\widehat{MTBFA} = \frac{\sum_{j=1}^{N} z_j}{\sum_{j=1}^{N} d_j} \tag{2}$$

where N is the number of independent experiments, d_j is a binary value for each experiment indicating whether or not a change was detected in the pre-change regime, $d_j \in \{0, 1\}$ and $z_j = min(d_d(j), d_s)$.

The Average Detection Delay (ADD) is evaluated under post-change regime to quantify the speed of a correct detection:[11] $ADD = \mathbb{E}_1(d_d)$, where the subscript 1 indicates the post-change regime. The estimate of ADD[10] where each experiment consists of n observations with a change-point d_s with $d_s < n$ is given as:

$$\widehat{ADD} = \frac{\sum_{j=1}^{N} z_j - d_s}{\sum_{j=1}^{N} c_j} \tag{3}$$

where c_j is a binary value for each experiment indicating whether or not a change was detected in the post change regime, $c_j \in \{0, 1\}$ and $z_j = min(d_d(j), n)$.

The term ARL (Average Run Length) is used in a sequential framework to unify MTBFA and ADD. It defines the MTBFA under pre-change regime, and the ADD under post-change regime.[10] The value of ARL can be theoretically derived given the CUSUM parameters, k and h and the pre- and post-change probability distribution of x_i. ARL can be approximated using a system of linear algebraic equations when x_i follows a Gaussian distribution.[12]

2.3 Datasets

A sudden or abrupt drift was simulated using three approaches: (i) a numerical simulation to detect the change in the mean of Gaussian data, (ii) A numerical simulation that altered the distribution of the data that is used as input to a neural network classifier, and (iii) a drift simulation where the input images to an AI CAD model are filtered using a median filter to induce a change in the sensitivity and specificity of the model.

2.3.1 Change in the mean of Gaussian data

Many performance measures, such as sensitivity and specificity, calculated over reasonably-sized populations, can be approximated by Gaussian distributions. To study the drift in such a performance metric, data was sampled from two univariate Gaussian distributions that differed in their means to represent the pre- and post-change regimes. We simulated a pre-change mean of 0.86, a post-change mean of 0.83, and a standard deviation of 0.05. We ran the simulations for a minimum of 1000 pre- and 1000 post-change days. To calculate ADD without censoring, some of the experiments were run for a larger number of days depending on the combination of the normalized threshold and the normalized reference value. We ran 1,000 experiments to estimate MTBFA and ADD empirically, and compared the ADD to ARL$_1$ (ARL in the post-change regime) calculated theoretically.

2.3.2 Change in the AUC of a two-class classifier applied to Gaussian data

We used two-dimensional Gaussian distributions to sample training data that belongs to two classes in the pre-change regime. The mean and the covariance matrix were used to define two statistically distinct data distributions. A Multi Layer Perceptron (MLP) classifier[13] was trained with samples from this training data, and tested during the pre-change regime. At the change-point, the means and co-variances of the two classes were modified to simulate a sudden data drift. The pre- and post-change average AUC values were 0.86 and 0.83 respectively, with a pre-change standard deviation of 0.05.

2.3.3 Change in the Specificity of an AI model for Patient Breast Imaging Data

We used the Emory Breast Imaging Dataset (EMBED) to detect a simulated drift in the specificity of an AI model for cancer detection on mammograms. EMBED[14] consists of 3.4 million two-dimensional (2D) and digital breast tomosynthesis (DBT) screening and diagnostic mammographic exams from 116,000 racially diverse patients from four hospitals over an 8-year period. Twenty percent of the EMBED dataset is available through Amazon Web Services (AWS) Open Data release. Despite the large size of the EMBED dataset, the number of cancer-positive cases in the screening dataset is limited due to the low prevalence of cancer in the screening population. In order to include an adequate number of independent cases in our simulation, we selected to monitor AI model specificity. We used 2D screening mammograms that are known to be negative for cancer, which included BIRADS 1 (negative) and BIRADS 2 (benign) cases, comprising of 12,800 patients with 12,800 exams and 61,298 mammograms.

The state-of-the-art Radiological Society of North America (RSNA) Screening Mammography Breast Cancer Detection AI challenge winning AI model[15,16] was deployed on the EMBED data. Changes in the model performance were monitored by measuring the performance metric, specificity, before and after inducing drift.

The drift was introduced by employing a median filter to denoise the images after the change-point day. For 60 pre-change days, 100 original exams per day were deployed on the RSNA model. The specificity of the model on the original mammograms was 0.837, which meant that the in-control mean was μ_0=0.837. The change-point was introduced at the end of the pre-change period with the denoised mammograms (out-of-control specificity, $\mu_1 = 0.876$) for a length of 60 post-change days. Thus, one experiment consisted of a series of 120 days of observations. By randomly choosing different sets of 100 exams per day, we performed a total of 1,000 experiments in our simulation. One cancer-negative exam (oldest) per patient was used in the 120-day period.

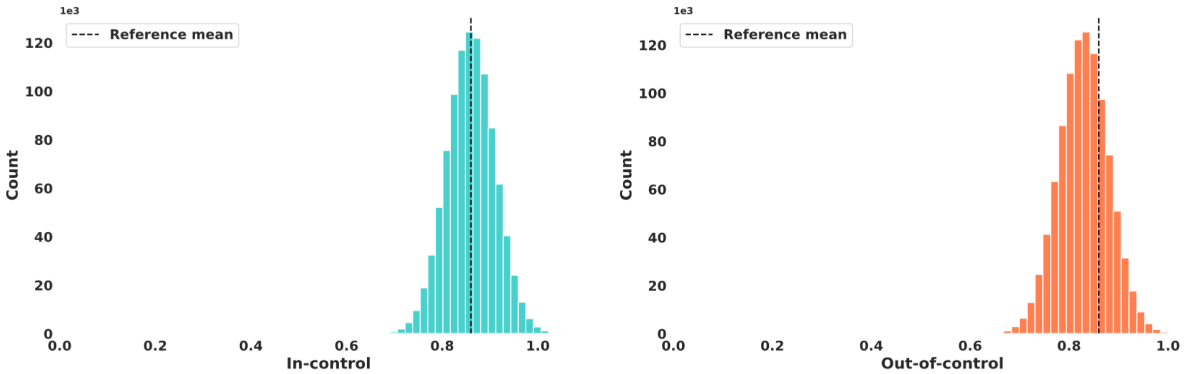

Figure 1: Gaussian data distribution before (*left*) and after (*right*) change-point.

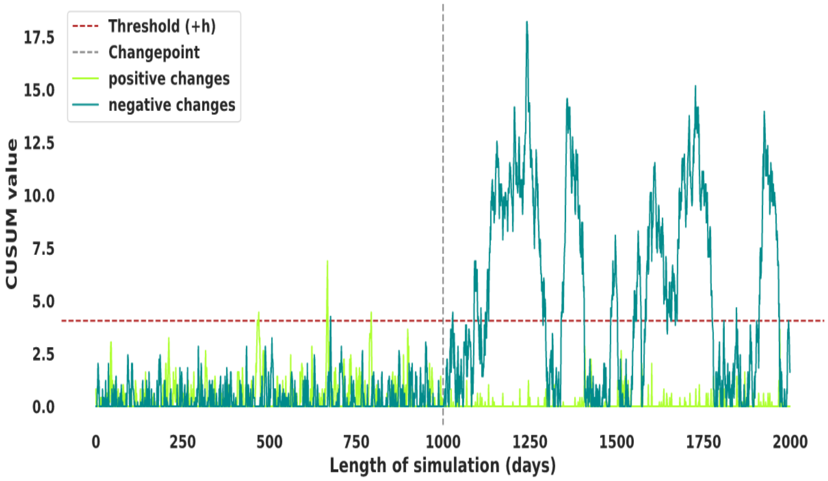

Figure 2: CUSUM control chart to detect the change (decrease) in the mean of a Gaussian random variable by monitoring the S_{hi}, S_{lo} for $h = 4$ and $k = 0.6$. y-axis shows the chart statistics S_{hi} and S_{lo} calculated using Equation (1) when the input data into CUSUM is normalized to have unity standard deviation.

3. RESULTS

3.1 Change in the mean of Gaussian data

The changes in the mean of the Gaussian data in the pre- and post-regime (Fig. 1) and the change-point detection using the CUSUM chart is illustrated in Fig. 2. S_{lo} is more relevant in this experiment since we are simulating a drop in the Gaussian mean from 0.86 to 0.83. In this example of the CUSUM chart with $h = 4$ and $k = 0.6$ for one of the experiments, we observe several false positives prior to the change-point and the first true detection within 22 days after the change-point. We detect the first false positive and true detection in each experiment to calculate MTBFA and ADD.

The results for change-point detection of the mean of a Gaussian random variable are summarized in Table 1. The average MTBFA and ADD values are obtained for a series of 1000 experiments. The detection delays are lower when CUSUM parameters are selected as $h = 4$ and $k = 0.6$, and this parameter choice also produces the most false alarms with the lowest MTBFA of 404.44. Higher values of MTBFA indicate fewer false alarms. As described Section 2.2.1, ARL can be approximated using a system of linear algebraic equations when the data is Gaussian. We used this approach to estimate a theoretical value for ARL_1 given h, k, the change δ in the mean (0.03 in our simulations) and the standard deviation σ (0.05 in our simulations). Theoretical ARL_1 was calculated using SPC R package function xcusum.ad,[17] which computes the steady-state ARL, defined as the ARL where the change appears after the process has been operating in steady state. The ADD from the simulation and the predicted ARL_1 from theory differ by less than 4%. For larger threshold and drift values, the simulation days were extended to detect the change-points, so that ADD can be calculated without censoring. For higher values of the reference value and threshold, additional observations are required for the cumulative sums to catch up to the detection threshold, and therefore, the detection delays are larger.

Table 1: Average detection delay (from simulations) and ARL$_1$ (from theory) to detect a change in the mean of a Gaussian random variable from 0.86 to 0.83 with a standard deviation of 0.05.

Simulation days	**Normalized Threshold** h	**Normalized Reference Value** k	**Average Detection Delay** (Observed)	**ARL$_1$** (from theory)	**MTBFA** (Observed)
1000	4	0.6	26.5	25.66	404.44
1000	5	0.6	37.24	36.88	1250.7
1480	4	1	184.0	176.5	8197
3200	5	1	423.55	407.84	62518.79

Table 2: Average detection delay (from simulations) and ARL$_1$ (from theory) to detect a change in the AUC of an ML classifier from 0.86 to 0.83 with a standard deviation of 0.05.

Simulation days	**Normalized Threshold** h	**Normalized Reference Value** k	**Average Detection Delay** (Observed)	**ARL$_1$** (from theory)	**MTBFA** (Observed)
1000	4	0.6	25.7	25.66	468.44
1000	5	0.6	39.4	36.88	473.9
1000	4	1	159.88	176.5	1037.8
2000	5	1	387.4	407.84	9935

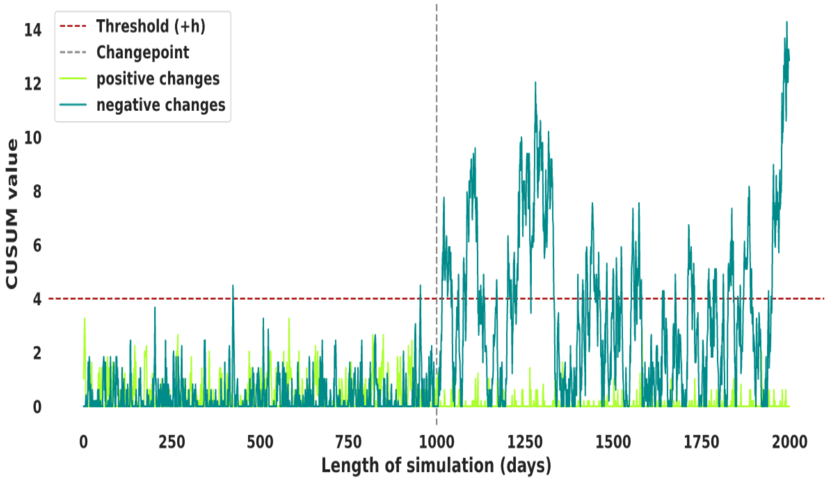

Figure 3: CUSUM control chart to detect the change in the classifier AUC by monitoring the S_{hi}, S_{lo} for $h = 4$ and $k = 0.6$. y-axis shows the chart statistics S_{hi} and S_{lo} calculated using Equation (1) when the in-control performance measure AUC is normalized to have unity standard deviation.

3.2 Change in the AUC of a two-class classifier applied to Gaussian data

CUSUM chart to detect changes in the AUC of a two-class classifier applied to Gaussian data by monitoring S_{hi}, S_{lo} for one of the experiments where CUSUM parameters were selected as $h = 4$ and $k = 0.6$ is shown in Fig. 3. The CUSUM chart shows the false positives and true detections while monitoring a drop in the classifier mean AUC from 0.86 to 0.83 where the change was introduced on day 1000. This figure shows that the first false positive was produced on day 459 and the first true-positive was produced on day 1027, incurring a detection delay of 27 days.

The results for detecting a change in the performance (AUC) are summarized in Table 2. Based on 1000 experiments, we obtained MTBFA and ADD values as shown in the table. The changes are detected faster for $k = 0.6$ compared to when $k = 1$. However, for $k = 0.6$, we also observed a larger number of false-positives, with a mean time between false-positive alarms of 468.4 and 473.9 for $h = 4$ and 5 respectively. Since there were fewer false positives with $k = 1$ in the pre-change period, the MTBFA is higher than with $k = 1$. As shown in Table 2, the average values for ADD based on the experiments are close (within 10%) to the ARL_1 values obtained using a Gaussian approximation to the data.

3.3 Change in the Specificity of an AI model with EMBED data

The sudden drift simulation using EMBED data consists of a series of observations spanning 120 days, which were obtained for over a length of 60 pre- and post-change days as shown in Fig. 4. Changes in the distribution of the AI model specificity due to the variation in the image characteristics as a result of denoising is depicted in Fig. 5. The mean specificity of the AI model prior to denoising was 0.837. After denoising with the median filter with $k = 10$, the model mean specificity increased to 0.876. To detect the changes in the AI model specificity using CUSUM charts, S_{hi}, S_{lo} and CUSUM of mean specificity are monitored as shown in Fig. 6.

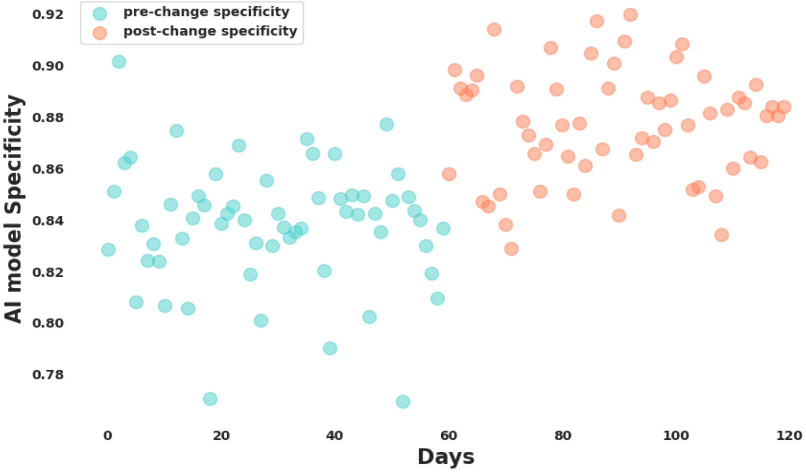

Figure 4: Sudden drift simulation with EMBED data: pre- and post-change specificity of the AI model.

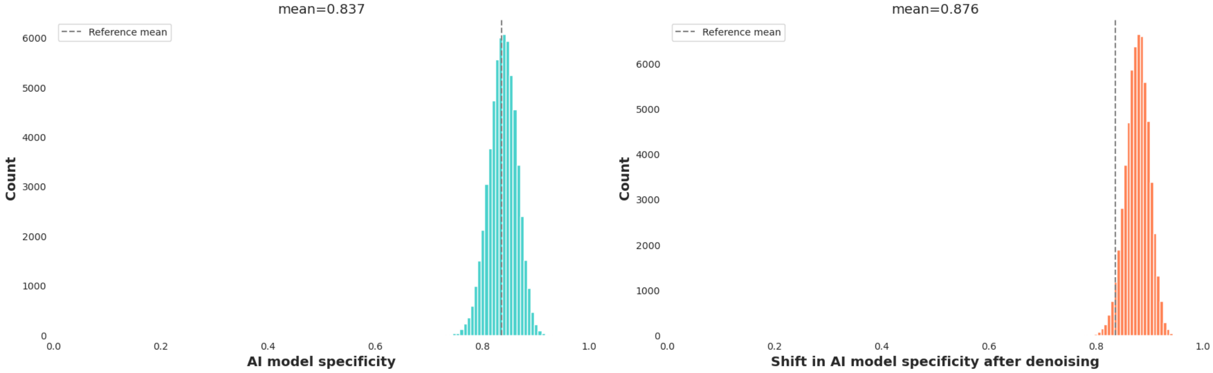

Figure 5: AI model specificity before (*left*) and after denoising (*right*) the input mammograms to simulate a performance drift.

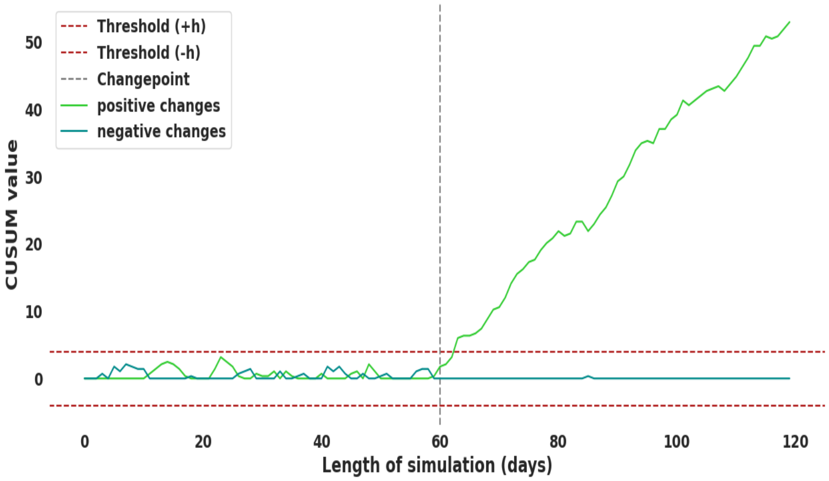

Figure 6: CUSUM control chart for monitoring the changes in the AI model specificity where the y-axis shows the chart statistics S_{hi} and S_{lo} calculated using Equation (1) for $h = 4$ and $k = 0.5$. AI model specificity is normalized to have a unity standard deviation.

Table 3: Average detection delay (from simulations) and ARL_1 (from theory) to detect a change in the specificity of an AI model from 0.837 to 0.876 with a standard deviation of 0.025 using the screening mammograms from the EMBED dataset.

Simulation days	**Normalized Threshold** h	**Normalized Reference Value** k	**Average Detection Delay** (Observed)	$\mathbf{ARL_1}$ (from theory)	**MTBFA** (Observed)
60	4	0.5	3.14	4.33	284.88
60	5	0.5	4.12	5.29	989.91
60	4	1	7.06	8.22	14979
60	5	1	8.9	10.21	nan

The results for change-point detection using EMBED data for 1000 experiments are summarized in Table 3. The detection delay is computed for a range of normalized threshold and reference values to detect a change of 1 and 2 standard deviations i.e. $k \in \{0.5, 1\}$. The changes in the mean specificity are detected more quickly for lower threshold and reference values. For $k = 0.5$ and $h = 5$, the performance change was detected in an average of about 4 days. With $k = 1$, the average detection delay increased by up to 5 days (e.g., from 4.12 to 8.9 for $h = 5$). The MTBFA was undefined for h=5 and k=1 because no false positives were identified in any of the experiments we conducted under this condition. The theoretical ARL_1 values provided a reasonable estimate of what we should expect in terms of average detection delay, although the observed ADD was less than ARL_1 under all conditions studied.

4. DISCUSSION AND CONCLUSION

In this work, we address the fundamental problem of detecting a change in the performance of an AI model during clinical use, which typically results from data drifts. We developed and tested a technique to detect performance drift of an AI model using CUSUM charts. We experimentally studied change-point detection using three different datasets, two of which used simulated numerical data and one used screening mammograms from the EMBED dataset. We investigated the use of CUSUM charts for change detection, and studied trade-offs among how quickly one can detect a change, the magnitude of the change with respect to inherent variability, and how fast a false-alarm is produced for a performance change. We simulated various conditions with and without change, including an example where a clinical site might start applying a denoising filter to mammograms before the data is presented to an AI model.

We demonstrated a data drift detection mechanism by identifying a change in AI model performance through monitoring metrics such as AUC and specificity. Our results indicate that with the appropriate choice of parameters, CUSUM is able to quickly detect relatively small drifts in performance with a large mean time between false-alarms. A limitation of CUSUM is that this method needs a set of cases initially to establish a target mean and an in-control standard deviation to then compute the deviations from the target mean. Therefore, to implement this in a clinical setting, there has to be an initialization period where the system is in-control to establish the target mean and the in-control standard deviation. Detection of performance change is the first step in identifying a mismatch between training/initial testing conditions and clinical use, which should lead to corrective action such as alerting the users, identifying and correcting the cause of the drift, model recalibration, or model retraining.

Acknowledgments

This project was supported in part by an appointment to the ORISE Research Participation Program at the Center for Devices and Radiological Health (CDRH), U.S. Food and Drug Administration, administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and FDA. The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health and Human Services. This is a contribution of the U.S. FDA and is not subject to copyright.

REFERENCES

[1] Moreno-Torres, J. G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N., and Herrera, F., "A unifying view on dataset shift in classification," *Pattern Recognit.* **45**, 521–530 (2012).

[2] Sahiner, B., Chen, W., Samala, R. K., and Petrick, N., "Data drift in medical machine learning: implications and potential remedies," *The British Journal of Radiology* , 20220878 (2023).

[3] Page, E. S., "Continuous inspection schemes," *Biometrika* **41**(1/2), 100–115 (1954).

[4] Fretheim, A. and Tomic, O., "Statistical process control and interrupted time series: a golden opportunity for impact evaluation in quality improvement," *BMJ quality & safety* **24**(12), 748–752 (2015).

[5] Novoa, N. M. and Varela, G., "Monitoring surgical quality: the cumulative sum (cusum) approach," *Mediastinum* **4** (2020).

[6] Woodall, W. H. and Adams, B. M., "The statistical design of CUSUM charts," *Quality Engineering* **5**(4), 559–570 (1993).

[7] Montgomery, D. C., [*Introduction to statistical quality control*], John wiley & sons (2019).

[8] Page, E. S., "Continuous inspection schemes," *Biometrika* **41**(1/2), 100–115 (1954).

[9] Lorden, G., "Procedures for reacting to a change in distribution," *The annals of mathematical statistics* , 1897–1908 (1971).

[10] Sahki, N., Gégout-Petit, A., and Wantz-Mézières, S., "Performance study of change-point detection thresholds for cumulative sum statistic in a sequential context," *Quality and Reliability Engineering International* **36**(8), 2699–2719 (2020).

[11] Pollak, M., "Optimal detection of a change in distribution," *Ann. Statist.* **13**(1), 206–227, year=1985.

[12] Goel, A. L. and Wu, S., "Determination of ARL and a contour nomogram for cusum charts to control normal mean," *Technometrics* **13**(2), 221–230 (1971).

[13] Taud, H. and Mas, J.-F., "Multilayer perceptron (MLP)," (2018).

[14] Jeong, J. J., Vey, B. L., Bhimireddy, A., Kim, T., Santos, T., Correa, R., Dutt, R., Mosunjac, M., Oprea-Ilies, G., Smith, G., et al., "The emory breast imaging dataset (embed): A racially diverse, granular dataset of 3.4 million screening and diagnostic mammographic images," *Radiology: Artificial Intelligence* **5**(1), e220047 (2023).

[15] Stephens, K., "RSNA announces screening mammography AI challenge results," *AXIS Imaging News* (May 10 2023).

[16] "RSNA Screening Mammography Breast Cancer Detection 1st place solution." Online https://www.kaggle.com/competitions/rsna-breast-cancer-detection/discussion/392449. (Accessed: 6 August 2023).

[17] "Statistical Process Control – Calculation of ARL and Other Control Chart Performance Measures." Online https://rdocumentation.org/packages/spc/versions/0.6.7. (Accessed: 15 August 2023).