# Change point Detection in Periodic Spatio-Temporal Hotspot

Mohd Sadique Raheem Shaikh
2021aim1012@iitrpr.ac.in
Venkata M. V. Gunturi
gunturi@iitrpr.ac.in
Department of Computer Science and Engineering, IIT Ropar, Rupnagar, Punjab, India

**Abstract**

The changes in a spatiotemporal periodic pattern can be due to various reasons like some special event(holidays) which causes the traffic flow to increase in the cloud temporarily, or the diversion of traffic from one city to another due to some reasons(i.e. natural calamity, a breakout of war, etc.). These changes can be an anomaly point which occurred temporarily or a change in a pattern which persisted throughout the pattern. These both can be detected using the likelihood function in the supervised learning method. We focus on the persistent change detection of the pattern and the change point at which it occurred. The main aim is to detect the change throughout the pattern and not an anomaly point. We are using the CUSUM technique for changepoint detection. The code implements Naive CUSUM and Modified CUSUM algorithms for detecting changes in a time series. The current techniques are for non-periodic patterns, which do not apply to PSTH patterns. The input data is split into timeslots, and for each timeslot, we use the Modified CUSUM algorithm and present appropriate experiments and results.

## 1 Introduction

The problem of change point detection in periodic Spatio-temporal hotspots (PST-Hotspot) takes the following inputs: (1) a Spatio-temporal event framework **E** (2) a set of PST-Hotspot for **E** where each hotspot **H** has properties (a) periodicity value(T), mean($\mu$), variance($\sigma^2$), change threshold($\theta$). Based on this input, the Monitor System **M** will initialise other parameters to each hotspot, such as sliding window size(W), change in mean value observation($\delta$), deviation from expected mean value(s), Cumulative sum of deviation(S) and decision function(GX). M will constantly monitor the PSTH patterns and alarm if the change exceeds the threshold for any hotspot. Figure 1 illustrates the graphical representation of the Monitor system M. where F(x) is the real-time data monitored and E is the framework input for initialisation.

### 1.1 Application Domain

The Change Hotspot Detection finds its application in optimising cloud services for cloud infrastructure companies like Microsoft. Cloud services are used by various entities, from large organisations to single users, so the requirements differ for different user groups. If

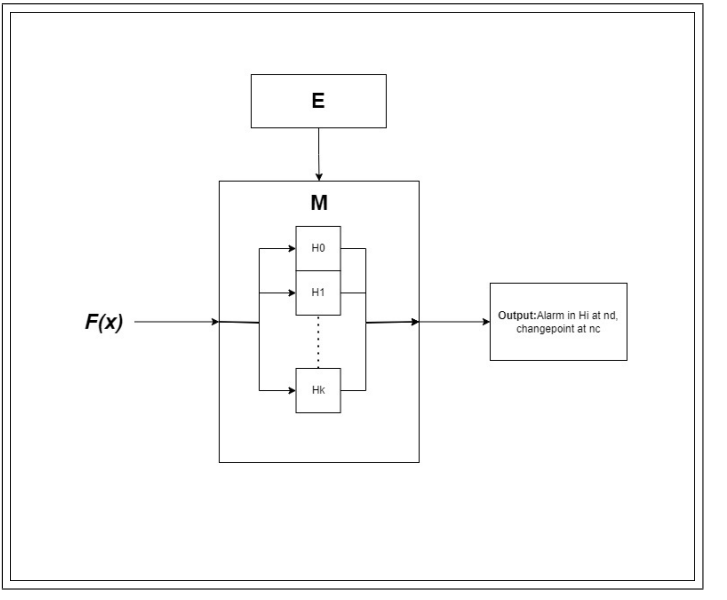


Figure 1: Monitor System for PST-Hotspot

in case of due to some unknown factor, the traffic load for one server may increase, For example, a "Significant amount of traffic from Helsinki suddenly starts to see very high RTT values which were not present a few days back." Such changes could primarily result from unexpected events such as the closure of Autonomous Systems (due to a sudden break out of the war), transoceanic fibre cut, etc. We plan to investigate the following approach for the change detection problem. We would develop a monitoring system to observe the patterns in real-time. Use the CUSUM technique with a defined set threshold for the change determined as valid—furthermore, the change point from where it occurred.

**Azure Traffic Dataset:** The dataset provided by Microsoft's WAN team contains information on sessions where each row entry has information about the Source IP address, timestamp, destination server, Round trip time(RTT) and a heavy bandwidth flag. This data covers cities such as Pune, Delhi, Bangalore, etc. Except for Microsoft data, we also use synthetic datasets with similar properties, such as PSTH, for experimentation. In Microsoft data, the change is not present initially. We use fault injection for further data generation based on current with similar data properties. The fault-injected data will contain the change point detected by the Monitor system.

## 1.2 Challenges

**1)Differentiating anomaly peak points from pattern change:** The anomaly peak point is a point which has a spike on it, i.e. the change in value is significant compared to its neighbours. , Otherwise, the pattern is followed before and after it. Unlike the change, we want to detect where all the points after the changepoint have a shift in average mean value. The Microsoft Azure Dataset contains peak points contributing to the PSTH pattern's periodicity.

For the likelihood ratio-based approach, the points give false alarms for the algorithm used in the change finder, such as AutoRegressor models. The challenge is setting the threshold value for the decision function so that the anomaly points change should not impact the decision and give a false alarm. The CUSUM techniques are generally immune to these changes, but if the magnitude of change concerning ongoing patterns is very high, it may trigger a false alarm.

**2) Trade-off between a false alarm and detection delay:** There exists a trade-off between change detection and the frequency of false alarms. If the change point is declared very quickly, there is a chance that it is a false alarm. However, if the change declaration delay is significant, the importance of change detection reduces, which may compromise the timely and valuable decision-making based on the detected change. The objective is to find an optimal trade-off between these two parameters. The aim is to achieve timely and accurate change detection while minimising the occurrence of false alarms, ensuring that the identified changes are meaningful and actionable for effective decision-making processes.

## 1.3 Limitations of related work

In the Literature survey [1], the comparison of a different existing algorithm for change detection for time series data is made on various scales like online detection by the algorithm, scalability, Algorithmic Constraints, efficiency, and performance evaluation. [3] also provides in-depth knowledge of different techniques used for change detection. However, the existing state-of-the-art focuses only on the time change, which is non-periodic, and there needs to be present work for detecting the change of periodicity of the hotspots as the PST-Hotspot is a new concept. We have proposed the Modified CUSUM, which will help in the change detection of PSTH patterns. As the CUSUM technique comes under the category of likelihood ratio methods, the Autoregressor methods also come under this category, used to detect the anomaly of a single-point change like [2]. [8] proposed the non-parametric method for change detection based on the ratio of probability densities. The CUSUM technique should be able to filter out anomaly point detection from the desired change point. The [9] provided the original and most basic CUSUM algorithm. As it is the most basic, it only satisfies the requirements for some cases. So the other CUSUM techniques are adapted version of it. The review of the CUSUM algorithm by [5] The CUSUM algorithm - a small review gives an overview of the CUSUM and also defines the terms involved in it briefly. For multiple dimension changepoint detection [4] proposed the CUSUM technique using binary segmentation on multidimensional data, but it works if the data is correlated. Our approach to Modified CUSUM is motivated by The CUSUM technique with timeslot sequence [7] because it is the only approach which has been derived for the periodic samples and on the network surveillance data, which is closely related to our dataset but the experimentation on network traffic is performed. In contrast, we are applying the round-trip time.

**Outline:** The remaining paper is structured as follows. Section 2 provides a comprehensive overview of the basic concepts necessary for understanding the topic and presents a formal problem definition. In Section 3.1, we present our proposed CUSUM measure, introducing its methodology and explaining its significance in addressing the problem. Section 3.2 focuses on the Modified CUSUM, where we discuss the modifications and enhancements made to the original measure to improve its effectiveness. Section 4 presents the experimental results obtained from research, insights and analysis based on the conducted experiments. Finally, in Section 5, we conclude the paper by summarising the key findings, contributions, and potential avenues for future research and development.

# 2 BASIC CONCEPTS AND PROBLEM DEFINITION

**Definition: ST-framework (E): E** defines the Spatiotemporal boundaries of the input dataset. E is the collection of PST-Hostspots(H) and time horizons where each hotspot has unique properties. Here E is the adapted version from [2]
**Definition: Periodic Spatio-Temporal Hotspot(H):** A periodic spatiotemporal hotspot H is a spatial region that shows a high intensity of events during a specific time window where high intensity repeats itself with periodicity. The properties of hotspot that are useful for us are (a)Mean value of RTT($\mu$), (b) variance of RTT($\sigma^2$), (c) High-Intensity time window(T), (d) sliding window size(W) (e) the change in mean value to detect($\delta$) (f) statistics to calculate the deviation from the expected mean(s) (g) cumulative sum of deviation over time(S) (h) decision function based on 'S'—used for alarming if the value become equal or greater than the threshold. The decision function is always greater than or equal to 0.
**Definition: Hot-region Window(T):** The time window during which the PSTHotspot shows a high intensity than the other hours in a repetitive pattern.
**Definition: Sliding Window(W):** The window frame considers the most recent data of the pattern equal to the length of the window—the calculation of mean and variance using a sliding window in Modified CUSUM. The default value is set as 10.
**Definition: mean value change($\delta$):** the change in mean value in log-likelihood considered for the detection.
**Definition: Deviation from the expected mean(s):** statistics to calculate the deviation from the expected mean value. The difference between actual and expected mean.
**Definition: Cumulative Sum(S):** The cumulative sum of deviation from the expected mean(s) over the pattern iteration.
**Definition: Decision Function(GX):** decision function based on 'S'. The alarm is triggered when the value becomes greater than or equal to the threshold. The function is always greater than or equal to 0.
**Definition: Alarm Threshold($\theta$):** The Threshold value for which the decision function triggers the alarm for change detection.
**Definition: Changepoint (nc):** The point at which the change in the mean value occurred—the mean before and after significant changes.
**Definition: Detection Point(np):** The point at which the decision function exceeded the threshold value.

## 2.1 Problem Definition

**Input:**
a) An ST-event framework E to initialise the Monitor system. Where E is a collection of PST-Hotspots (H) and each Hotspots(H) have properties each hotspot **H** has properties (a) periodicity value(T), mean($\mu$), variance($\sigma^2$), change threshold($\theta$).
2)F(x) real-time value of x: where x is the RTT value feed to the Monitor system in real-time.

**Output:**
A collection of PST-Hotspots if the change in hotspot occurs; otherwise, NULL. The collection will contain Hotspots with change, and each hotspot will have :(a) changepoint(nc) and (b) detection point(nd).

# 3 PROPOSED APPROACH

The Solution to the problem of changepoint detection in PSTHotspot using the CUSUM technique. We propose two different CUSUM techniques 1) Naive CUSUM and 2) Modified CUSUM. Naive CUSUM is a derivation for simple CUSUM with the implementation of a decision function specific to our requirements of PSTHotspot. Modified CUSUM is a further improvement in the Naive method with respect to calculation optimisations, approximations, timeslot updates and sliding window insertion. We will first describe Naive CUSUM in section 3.1, then the Modified CUSUM in section 3.2.

## 3.1 Naive CUSUM

The key steps involved in the proposed Naive CUSUM approach are
**1.Initialisation:** Initialise the variables like $\delta$ that represents the value used in the CUSUM calculation, **S**: Cumulative sum of the CUSUM statistics, **GX:** Maximum cumulative sum of the CUSUM statistics, $\hat{\mu}$**:** Estimate of the mean, $\hat{\sigma}$**:** Estimate of the variance,**s:** CUSUM statistic. Set the initial values for $\hat{\mu}$ and $\hat{\sigma}$ based on the first data point.
**2.Naive CUSUM Calculation:** Iterate over each value x in the given data. Calculate the current estimates of the mean and variance using the previous data points and the current sample as follows:

$$\hat{\mu}_n = (\hat{\mu}_{n-1} * (n-1) + x_n)/n$$

where $\hat{\mu}_n$ is mean at point n and $\hat{\mu}_{n-1}$ is the mean at point n-1, and $x_n$ is rtt at point n

$$\hat{\sigma}_n = ((n-1) * \hat{\sigma}_{n-1} + (x - \hat{\mu}_n)^2)/n$$

where $\hat{\sigma}_n$ is variance at point n, $\hat{\sigma}_{n-1}$ is variance at point n-1.
Compute the CUSUM statistics (s) using the following equation:

$$s_n = \delta/\hat{\sigma}_n * (x_n - \hat{\mu}_n - (\delta/2))$$

Update the cumulative sum (S) and the maximum sum (GX) as follows:

$$S_n = S_{n-1} + s_n$$

where $S_n$ represents the cumulative sum at point n

$$GX_n = sup(GX_{n-1} + s_n, 0)$$

where sup() represents the supremum and assigns the largest value.
**3. Alarm Condition and Change Point Detection:** Check for an alarm condition by comparing the maximum cumulative sum (GX) with a predefined threshold value. If GX exceeds the threshold, it indicates an anomaly. If the alarm condition is triggered, indicating an anomaly, we identify the corresponding time slot (nd) as n and determine the nearest change point (nc) by finding the index with the minimum cumulative sum (S) among the available data points up to n.

The proposed Naive CUSUM approach is effective for the detection of changes in the data. The CUSUM statistic is reliable for identifying significant deviations from the expected behaviour. These calculations enable the timely identification of anomalies, facilitating further analysis and decision-making processes. However, calculating mean and variance estimates at every point x are time-consuming as after scanning a long pattern, the changes become nearly insignificant for every new point. The proposed Modified CUSUM reduces the computational cost with appropriate approximations.

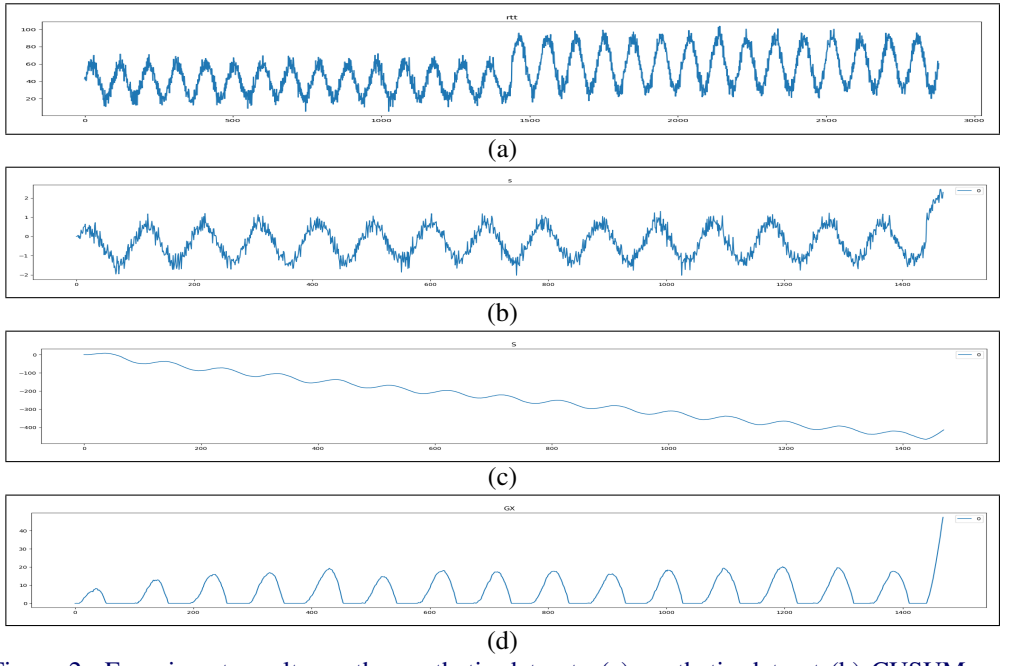


Figure 2: Experiment results on the synthetic dataset: (a) synthetic dataset (b) CUSUM statistics s (c) CUSUM function S (d) Decision function GX

## 3.2 Modified CUSUM

We introduce a Modified CUSUM method for hotspot detection in this proposed approach. The method utilises a sliding window approach to monitor data and detect anomalies. Described below are the main steps involved in the approach.

**1. Initialisation:**Initialise the parameters required for the hotspot detection, including the previous data, $\mu$, $\sigma^2$, hot region range(T), sliding window size(W), $\delta$, and mean estimate ($\hat{\mu}$), variance estimate ($\hat{\sigma}$), CUSUM statistics (s), cumulative sum (S), and decision function (GX).

**2. Update Parameters:** When a new data chunk is available at the end of the hot region, update the parameters by concatenating the new data with the previous data. Calculate the updated mean and variance for the new window of data:

$$\mu = \sum_{w=1}^{w_n} [\sum_{j=1}^{T_n} (x_j)/T_l]/W_l$$

where the first summation is for all the hot regions in the sliding window, and the second summation represents the individual hot region points sum. and $T_l$ and $W_l$ represent the length of hotregion and sliding window.

$$\sigma^2 = \sum_{w=1}^{w_n} [\sum_{j=1}^{T_n} (x_j - \mu)^2/T_l]/W_l$$

where $\sigma^2$ represents the variance over all the points that lie in sliding window W.

**3. Modified CUSUM Calculation:**Iterate over the samples for each hotregion in the given

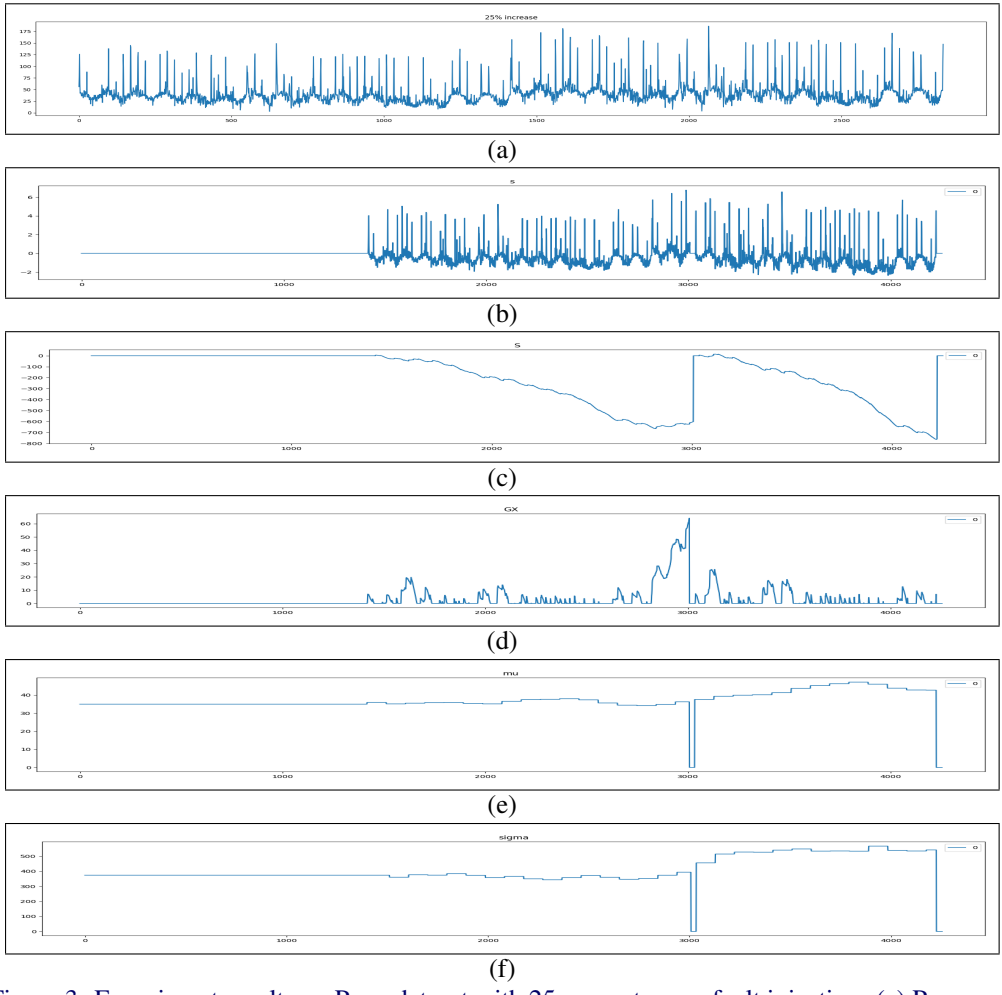


Figure 3: Experiment results on Pune dataset with 25 percent mean fault injection: (a) Pune dataset (b) CUSUM statistics s (c) CUSUM function S (d) Decision function GX (e) Mean value with sliding window size 10 (f) variance with sliding window size 10

data. Calculate the current estimates of the mean ($\hat{\mu}$) and variance ($\hat{\sigma}$) using the updated values. Compute the CUSUM statistic (s) based on the delta value, current sample, mean estimate, and adjusted variance estimate. Use the calculated CUSUM statistic to update the cumulative sum (S) and the maximum sum (GX). Check for an alarm condition by comparing the maximum sum (GX) with a predefined threshold. If the value exceeds the threshold, indicating an anomaly, record the corresponding time slot (nd) and determine the nearest change point (nc) by finding the index with the minimum cumulative sum (S) among the available data points up to that time slot.

**4.Output:**If an anomaly is detected, the identified time slot (nd) and the nearest change point (nc) are returned as the output. Otherwise, if no anomaly is detected, the output is set to None.

The proposed Modified CUSUM approach allows effective hotspot detection by con-

| Mean | Threshold | | | |
|---|---|---|---|---|
| (% increase) | 20 | 30 | 40 | 50 |
| 25% | 29 | 69 | 84 | 64 |
| 50% | 20 | 28 | 434 | 62 |
| 75% | 15 | 21 | 27 | 34 |
| 100% | 13 | 18 | 22 | 27 |

Table 1: Mean vs Trigger delay.

tinuously monitoring the data using a sliding window and identifying significant deviations from the expected mean values. The utilisation of mean and variance estimates, along with the CUSUM statistic, enables the timely detection of anomalies. The approach provides valuable insights for further analysis and decision-making in various applications.

# 4 Experiment and results:

The performed experiments use the Pune dataset to analyse different fault injections with mean increases ranging from 25% to 100%. The threshold value is also varied from a range of 10 to 50. The delta value also varied between 0.05 to 0.40 for the experiments.

**Mean increase vs trigger time:** In the experiment, we tried to investigate the relationship between the increased mean and trigger time of the CUSUM; the data was fault injected with increased mean of 25%, 50%, 75%, and 100% to observe the degree of mean increase and the time taken for alarm triggering. The experiment shows a clear correlation between the magnitude of the mean increase and the alarm trigger; as the mean increase, the alarm trigger is faster for anomalous behaviour. With the change point at observation 1323, the trigger alarm at 1417 for a 25% mean increase. Similarly, for mean increases of 50%, 75%, and 100%, the alarm trigger was observed at 1385, 1357, and 1350 at threshold value 20. With the increase in the threshold value, the time for the alarm trigger also increases. These highlight the effectiveness of a monitoring system; the ability to detect and address anomalies promptly enhances network security and performance, ultimately contributing to more efficient network management.

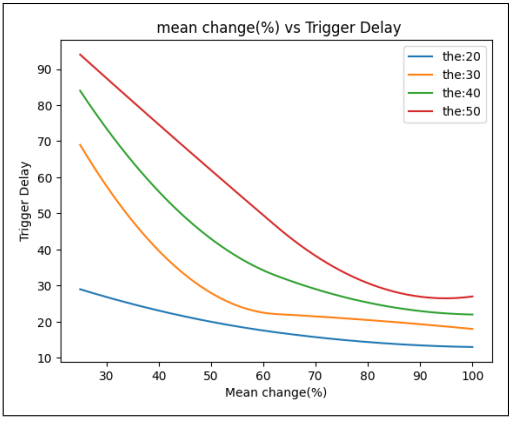


Figure 4: Mean vs trigger delay

**Threshold vs trigger time:** The experiment was performed on the Pune dataset with fault injection of a mean increase of 25% and a delta value of 0.05. The threshold value is varied in the range of 10 to 50. The experiment results show that threshold value significantly affects alarm trigger time. With a threshold value of 50, the alarm triggers at 1417, with a change point at 1323.

| Threshold | Delay |
|---|---|
| 50 | 94 |
| 40 | 84 |
| 30 | 69 |
| 20 | 29 |
| 17 | 25 |
| 16 | 24 |
| 15 | False Alarm |
| 10 | False Alarm |

Table 2: Threshold Vs Trigger delay

As the threshold value decreases, the alarm trigger time also decreases. It becomes more sensitive to change and responds quickly. For threshold values 40, 30, 20, and 17, the observed alarm trigger at 1407, 1392, 1352, and 1348 with a change point at 1323. However, it is essential to note that setting the threshold too low can trigger a false alarm; for threshold values 15 and 10, the false alarm trigger at 1182 and 101 occurred, which are even below the real change point. The experiment results indicate the trade-off between sensitivity and the risk of false detections when selecting the threshold value. Therefore the threshold value should be set carefully to the specific needs, balancing early anomaly detection and minimising false alarms.

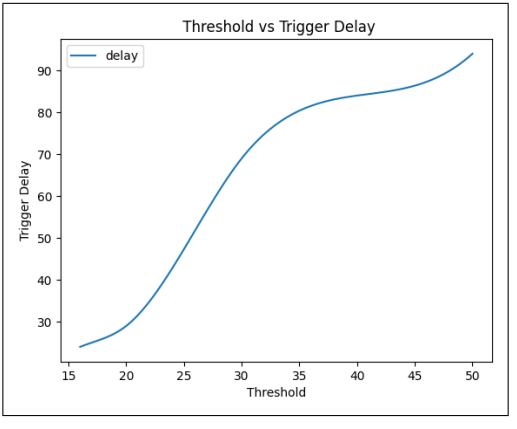


Figure 5: Threshold value vs trigger delay

**Delta value vs alarm trigger time:** Experiment performed on the Pune dataset with fault injection of 25% mean increase and set threshold value constant at 50. The delta value changed in the range of 0.05 to 0.40. The alarm trigger value is 1417, with the change point at 1323. As the delta value increases, the sensitivity to the delay time of the alarm trigger increases. For instance, with a delta value of 0.10, the alarm was triggered at observation 1400; for a delta value of 0.15, the alarm was triggered at 1404. The trend continued, and with delta values of 0.20 and 0.30, the alarm triggered at observations 1411 and 1420, respectively, with corresponding change points. Also, the change point value slightly increased as well. Interestingly, no detection occurred as the delta value increased beyond a certain threshold. In our experiment, when the delta value is set to 0.36 and 0.40, no alarm was triggered; as delta values indicate the minimum change required to trigger an alarm finding the right value is important

# 5 Conclusion:

The research paper focused on the implementation of a monitoring system for the data using Naive CUSUM and Modified CUSUM techniques. The goal is to detect the change in the

| Delta | nc | nc | Trigger Delay |
|---|---|---|---|
| 0.05 | 1323 | 1417 | 94 |
| 0.10 | 1324 | 1400 | 76 |
| 0.15 | 1329 | 1404 | 75 |
| 0.20 | 1329 | 1411 | 82 |
| 0.30 | 1402 | 1420 | 18 |
| 0.35 | 1404 | 1426 | 22 |
| 0.35 | Na | Na | Not Detected |
| 0.40 | Na | 1Na | Not Detected |

Table 3: delta vs Trigger delay.

mean value of a PST-Hotspot The conducted experiment on synthetic data and a real dataset of Pune with fault injection. The results showed that the alarm was triggered at different observation points depending on the input parameters, such as threshold and delta values. The values of the threshold and delta are directly correlated with the sensitivity, but if the threshold value is set too low, a false alarm starts triggering; if the delta value is set very high, the change might go undetected. Overall, the experimental results highlight the effectiveness of CUSUM techniques in detecting the changepoint. The threshold and delta parameters can be set based on the requirements. The research provides valuable insights for changepoint detection applications and lays the foundation for further advancement in detecting and analysing changepoints in real-time systems.

# References

[1] Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

[2] Alexander Aue, Siegfried Hörmann, Lajos Horváth, and Matthew Reimherr. Break detection in the covariance structure of multivariate time series models. 2009.

[3] Michele Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs, 1993.

[4] Haeran Cho and Piotr Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 475–507, 2015.

[5] Pierre Granjon. The cusum algorithm-a small review. 2013.

[6] Venkata MV Gunturi, Rakesh Rajeev, Vipul Bondre, Aaditya Barnwal, Samir Jain, Ashank Anshuman, and Manish Gupta. A case study on periodic spatio-temporal hotspot detection in azure traffic data. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1037–1044. IEEE, 2022.

[7] Daniel R Jeske, Veronica Montes De Oca, Wolfgang Bischoff, and Mazda Marvasti. Cusum techniques for timeslot sequences with applications to network surveillance. *Computational statistics & data analysis*, 53(12):4332–4344, 2009.

[8] Yoshinobu Kawahara and Masashi Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 389–400. SIAM, 2009.

[9] Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.