Lab 5: Detecting and Tracking Accidents on Google Maps

param parekh

November 2025

Motivation

In this Lab, we are going to study how to detect road accident occuring on busy roads and how long they last. We are going to study below three tasks:

1. Accident detection: determining whether an accident has truly occurred based on noisy user-submitted reports.
2. Accident persistence: once an accident is confirmed, monitoring traffic speed from GPS data to track the ongoing status of the event.
3. Accident clearance: identifying when traffic speeds return to normal, indicating that the accident has been cleared.

1 Accident Detection (Reports Coming In)

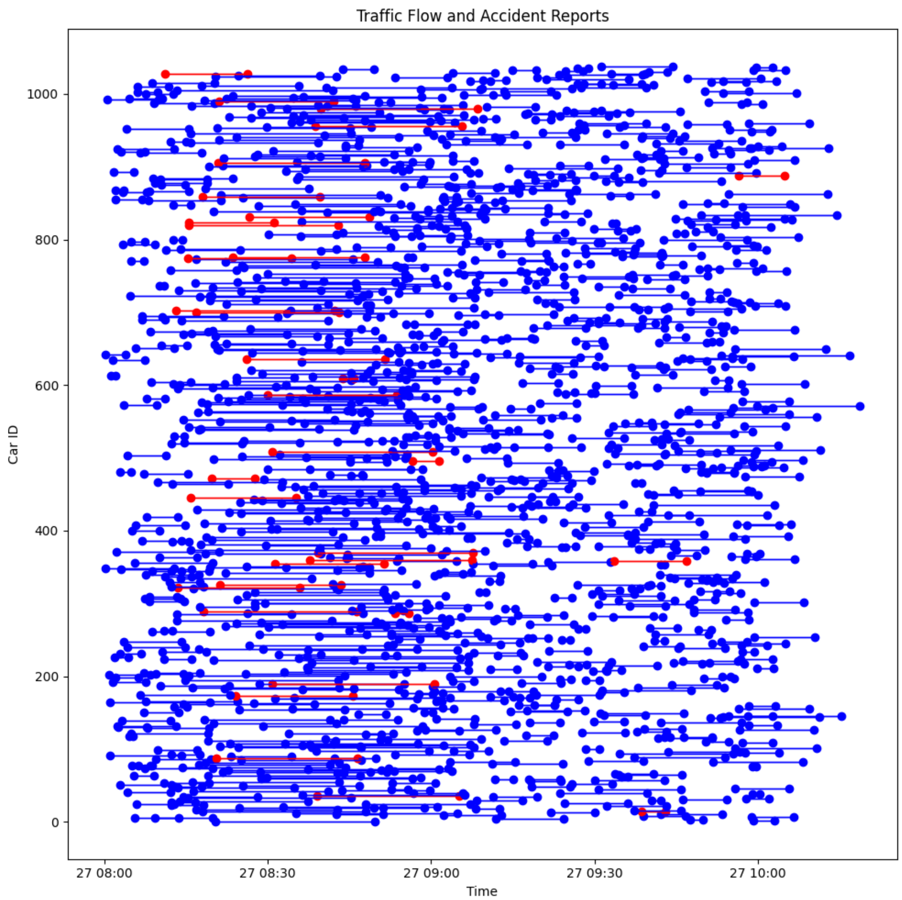
When an accident happens, platforms like Google Maps let users submit reports. However, these reports can be unreliable as some users might give false alerts, while others might not report at all. So, our main question is: "how confident can we be that an accident actually occurred?" To answer this, we'll use an inferential statistics method called hypothesis testing.

1.1 Load Data and Visualise

we visualized each car's movement between Point A and Point B, highlighting reported cars in red to identify when the accident likely started and when traffic returned to normal. Below plot depicts it.

1.2 Hypothesis Testing from Initial Reports

Using the Null hypothesis and the alternative hypothesis below, we can detect the accident in a particular time interval.
Null hypothesis H_0: No accident, in which case we may obtain a residual level



Observation: A cluster of red points appears around 08:15, indicating when accident reports began. By around 08:45, travel times shorten and reports stop, implying the accident was cleared.

Figure 1: Traffic Flow and accident reports

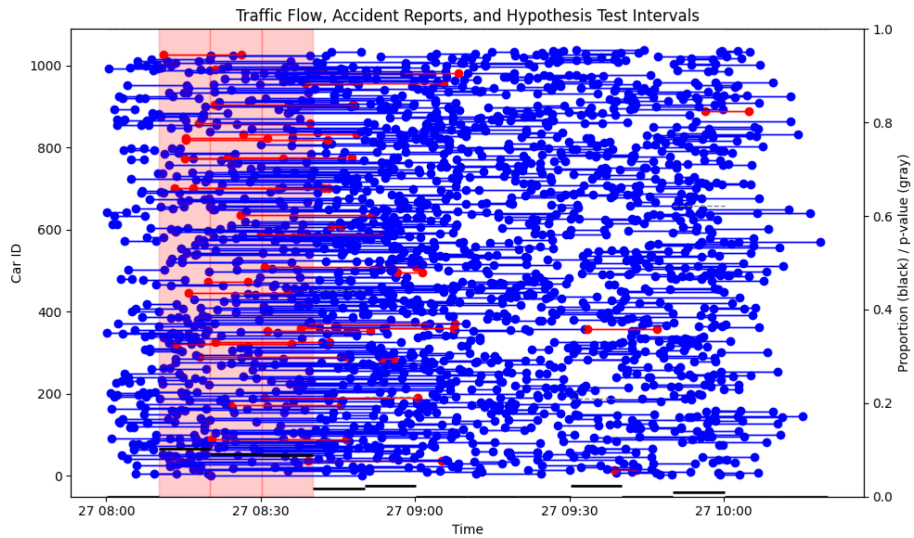


Figure 2: Hypothesis Testing from Initial Reports

of reports (1%).
Alternative hypothesis H_1: There is an accident, in which case accidents are reported at a rate of 10%.
Figure 2 contains results from this hypothesis testing.
Discussion Questions:

1. In what intervals did the hypothesis test reject H_0? Do these align with where you would visually expect accidents to occur?
   Red shaded intervals are those where the hypothesis test rejects H_0. according to figure 2, Intervals 8:10 to 8:35 rejects H_0, meaning accident is detected in those time intervals. This aligns with the visual expectation of accidents that we analysed in figure 1.

2. How does the number of cars n in an interval affect the reliability of the p-value?
   When the number of cars n in the interval increases, it produces small reporting rate r/n which leads to a stable p-value. When the number of cars n in the interval decreases, it procduces a relatively large reporting rate r/n which leads to small p-values, making the test more sensitive to noise.

3. What are the trade-offs of using a 90% confidence threshold instead of a stricter or looser one?
   Using a 90% Confidence Interval, we are willing to tolerate 10% of false positives in the test while rejecting H_0. If we decrease CI, it increases tolerance of false positives. If we raise the confidence level (e.g., to 95% or

99%), we decrease tolerance for false positives, reducing false alarms but making the system slower to detect real accidents.

4. Could random noise in reports cause false detections? How might you prevent this?
   Yes, if some cars randomly report false reports, if n is small, the reporting rate r/n might increase and produce low p-value showing artificially rejecting H_0. To prevent this, we can try out multiple ways described below:
   (a) Set minimum amount of cars in the interval to avoid unreliable p-value
   (b) Declare accident only if consecutive intervals reject H_0
   (c) Using smoothing techniques like moving averages to reduce random noise in the reporting rate

1.3 Hypothesis Testing from Travel Times

Since human reporting might be inconsistent, we used vehicle travel times as a second source of evidence to detect accidents. We define the following hypothesis.
Null hypothesis H_0: no accident. The travel times are normally distributed with a mean = 5 minutes and a variance = 1 minute.
Alternative hypothesis H_1: there is an accident. The travel times are normally distributed with a mean = 20 minutes and a variance = 5 minutes.

Result Plot 3 Analysis:

- We divided data into non-overlapping 10-minute interval blocks. The mean travel time of the block is displayed as black horizontal lines (Figure 3)
- For each interval, a likelihood-ratio test is performed to compare the null hypothesis of normal traffic conditions with the alternative hypothesis indicating an accident.
- Resulting p-values are plotted as gray dashed lines, and p-values fall below the 0.1 threshold are shaded in red to denote rejection of H_0 at the 90% confidence level.

Figure 3 shows the result of accident detection via likelihood ratio test intervals. The red shaded regions say that an accident occurred in those intervals.

Discussion Questions:

1. Which approach gives a clearer signal of accidents: reports or travel times?
   The travel time data gives clearer reporting signals of accidents. If we analyse figure 3, before 08:25, travel times cluster around the normal mean

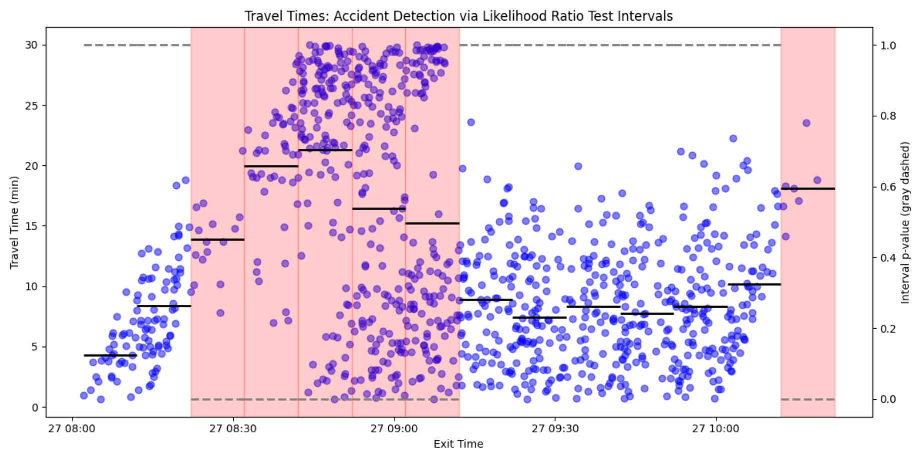


Figure 3: Hypothesis Testing from Travel Times

(5 minutes). Between 08:25 and 09:05, travel times jump sharply to nearly 20 minutes, and the interval averages rise. As soon as the accident clears, the travel times quickly fall back down toward normal levels. Hence, It is better, stronger detection policy. In contrast, the user-report-based detection was noisy and depended on inconsistent human behaviour.

2. What are the trade-offs of relying on human reports versus relying on sensor-based travel-time data for accident detection? How do these trade-offs relate to the hypotheses H_0 and H_1 used in this section?
There are pros and cons for both whether we rely on human reports or sensor-based travel-time data for accident detection. Human can report accidents early, but it can be unreliable wherease sensor-based travel-time are accurate but might be slower in intial response.
For human reports, the hypotheses $(p = 0.01)$ $vs.$ $(p = 0.10)$ are close, so random variation can easily declare an accident, making detection fast but noisy. For travel times, the distributions $(N(5, 1))$ $vs.$ $(N(20, 5))$ are far apart, giving a much stronger likelihood-ratio signal. This makes travel-time detection more reliable and stable.

2 Accident Persistence (Tracking Vehicle Speeds)

In Part 2, we shift from accident detection to accident persistence. We use GPS-based speed measurements that allow us to track whether traffic remains abnormally slow. Using sequential estimation, SPRT, and time-series forecasting models, we evaluate when speeds indicate that congestion is still ongoing and determine the point at which traffic has been normal to declare the accident cleared.

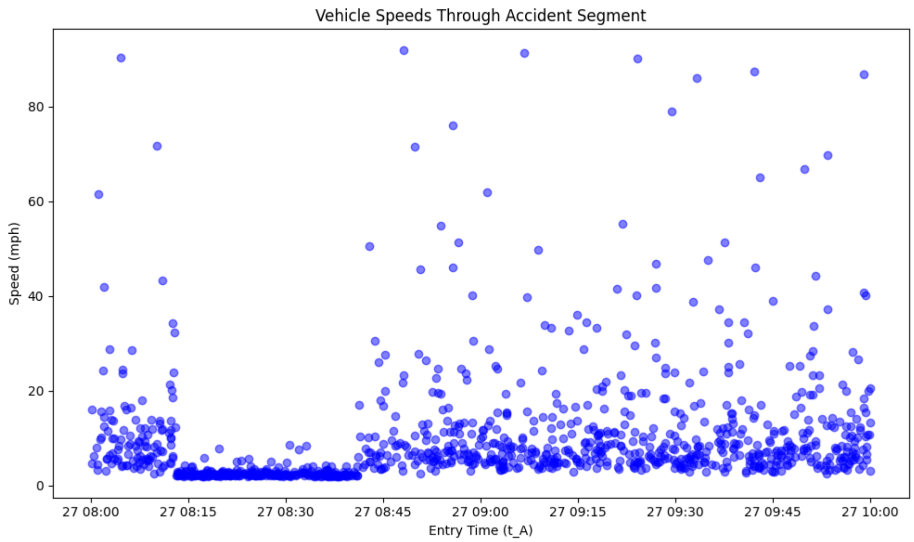


Figure 4: Vehicle speeds throughout intervals

2.1 Collect and Visualise Duration Data

Figure 4 shows the vehicle speed at entry point A. We can clearly see that around 8:15 to 8:45, speed is drastically low indicating a signal of an accident.

2.2 Sequential Estimation (Online Processing with Forgetting Factor)

As accident is detected, our system now wants to decide when to clear the accident alert, meaning is the traffic recovered enough to clear the accident alert? To achieve this, we can use sequential estimation techniques like the Exponential Moving Average to estimate mean travel duration in a noise-resistant, smooth way. The EMA updates the running estimate μ_t according to

$$\mu_t = (1 - \alpha)\, \mu_{t-1} + \alpha\, d_t,$$

where $0 \leq \alpha \leq 1$ is the forgetting factor. According to Figure 5,

- $\alpha = 0.5$: Red Line: Reacts very quickly to new values
- $\alpha = 0.1$: Blue Line: It is smooth enough to ignore noise
- $\alpha = 0$: Green Line: completely ignores new data

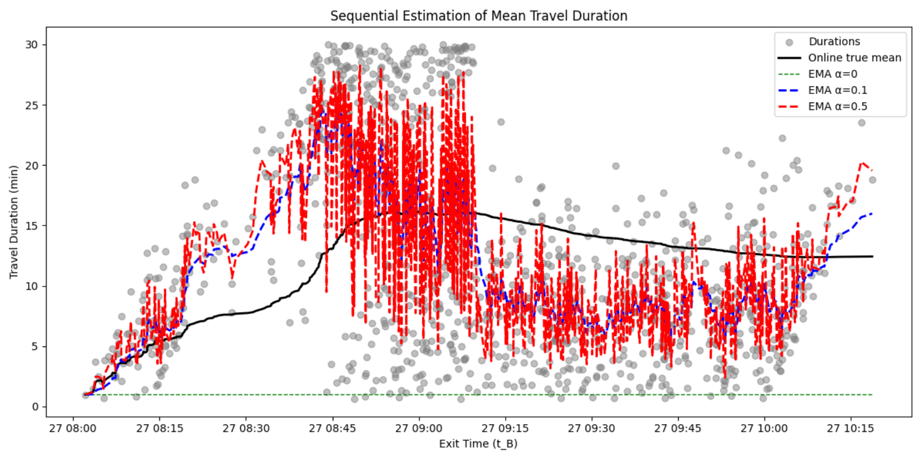


Figure 5: Sequential Estimation of Mean travel duration

2.3 Sequential Accident Detection with SPRT

In this step we apply the Sequential Probability Ratio Test (SPRT) to continuously evaluate whether traffic is currently in a *normal* state or an *accident* state. Let's assume each vehicle provides travel duration d_t under below two hypotheses:

$$H_0: \ d_t \sim \mathcal{N}(5,\, 2^2), \qquad H_1: \ d_t \sim \mathcal{N}(20,\, 5^2).$$

For every new observation, the SPRT updates a cumulative log-likelihood ratio (LLR),

$$\log LR_t = \log LR_{t-1} + \big[\log f_1(d_t) - \log f_0(d_t)\big],$$

which says how much more the evidence supports H_1 over H_0. At the end of each macro-interval, the test compares the LLR to two decision thresholds,

$$A = \frac{1-\beta}{\alpha}, \qquad B = \frac{\beta}{1-\alpha},$$

and uses their logarithms for computation.

- If $\log LR_t \geq \log A$, the system declares an *accident detected.*
- if $\log LR_t \leq \log B$, it declares *traffic cleared.*
- Otherwise, the test continues collecting data.

Figure 6 shows results from sequential accident detection with the SPRT technique.

Discussion Questions:

1. How quickly did the SPRT detect an accident after travel times increased?
   In my results, the SPRT first crosses the upper threshold and declares "accident detected" a short time after travel times start rising. Detection is not quick. The likelihood ratio needs several slow cars within one or two macro-intervals to accumulate enough evidence to exceed the upper threshold, but once durations cluster near 20–30 minutes, the LLR grows very rapidly.

2. After the accident cleared, how many cars (or how much time) did it take before the test crossed the lower threshold and declared "normal traffic"?
   After the green "accident cleared" line, travel times fall back toward the normal range. It takes several intervals of mostly normal-length trips before the cumulative LLR drops below the lower threshold and the SPRT formally declares "cleared".

3. Compare the trade-off between false alarms and missed detections: Which type of error (false alarm vs. missed accident) seems more serious in this application? How would adjusting α or β change the test's sensitivity?
   - In accident detection, a false alarm means the system incorrectly announces an accident, while a missed detection means it fails to warn drivers when an accident is real. In real traffic systems, missing a real accident is worse because unexpected delays can cause more traffic.
   - Smaller α, β make the thresholds stricter, so the system needs more evidence to make a decision, which reduces errors but increases detection delay.
   - Larger values make the system respond faster but also more likely to trigger false accidents.

4. Did you observe intervals where the likelihood ratio oscillated between the thresholds without reaching a decision?
   When the LLR keeps oscillating in the middle region (between decisions), it means the system is not fully sure whether the traffic is normal or still affected by the accident. Some cars may be slow, while others are fast, so the evidence is uncertain. So, SPRT waits for more cars instead of making a wrong decision too early. The oscillation basically shows that the data is uncertain, so the system keeps collecting more information.

5. In practice, what external signals (e.g., user reports, sensor data, time of day) could be combined with travel times to improve accident detection reliability?
   The system can use other signals along with travel times to improve accuracy. For example, human accident reports, sensor data showing lane closures, weather, and even historical accident patterns on that road. If multiple signals agree, the system can detect accidents faster and avoid false alarms.

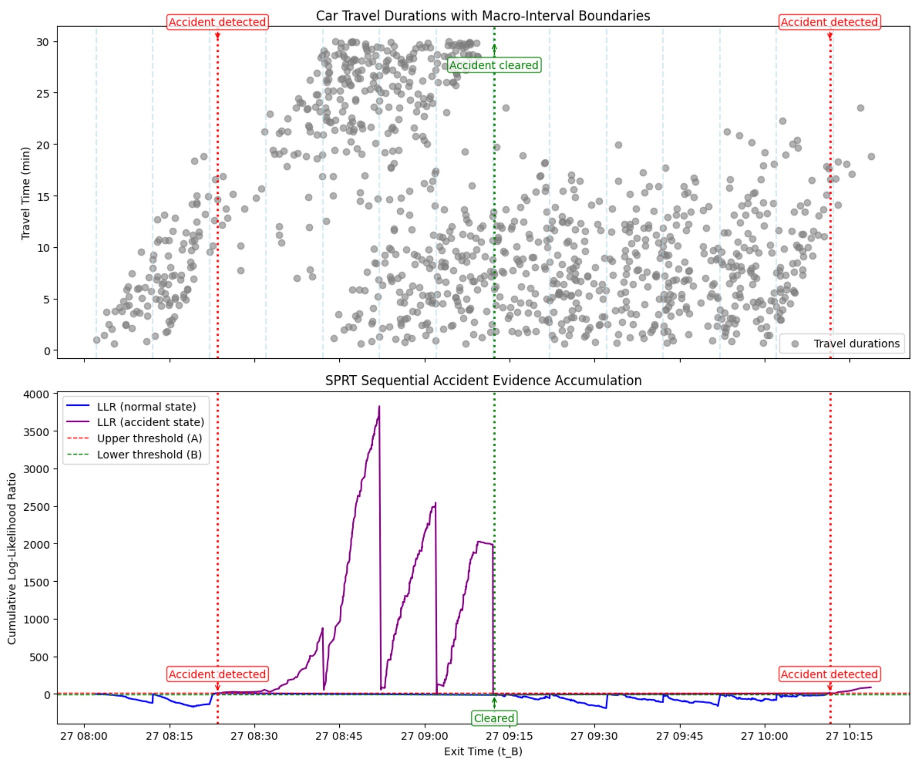


Figure 6: Sequential Accident Detection with SPRT

2.4 Rolling Forecasts with AR / ARMA / ARIMA

We extend the real-time tracking of accident persistence by using time-series models to forecast travel durations. At the end of each 10-minute interval, the system fits AR, ARMA, and ARIMA models to all observed durations up to that point and predicts the next 10 minutes of traffic conditions. This allows us to compare which model family is the most stable versus the most responsive estimate of future traffic recovery. We considered below models:

- AR (AutoRegressive): lags = 1, 2, 5
- ARMA (AutoRegressive + Moving Average): (2,1), (2,3)
- ARIMA (AutoRegressive + Integrated + Moving Average): (2,d,2) with d = 0, 1, 2

Discussion Questions

1. Do AR models with short vs. long lags respond differently to sudden drops?
   Short-lag AR models (e.g., AR(1)) react much faster to sudden decreases in travel times because they rely heavily on the most recent observation. Long-lag AR models (e.g., AR(5)) react more slowly since they average over a longer history. In Figure 7, the AR(1) forecasts drop sharply immediately after the accident clears, while AR(5) transitions more gradually. Thus, AR(1) is more reactive, while AR(5) produces smoother but slower adjustments.

2. Do ARMA models smooth out noise better than AR alone?
   ARMA models remove noise more effectively and produce smoother, more stable forecasts. The ARMA(2,1) and ARMA(2,3) forecasts appear as smooth, stable horizontal blocks inside each interval, without the fluctuations seen in AR models.

3. Does differencing in ARIMA (d = 0, 1, 2) make clearing forecasts more realistic, or does it over-smooth?
   Moderate differencing (d = 1) makes the forecasts more realistic, but too much differencing (d = 2) can over-smooth the forecasts, making them slow to return to normal even after travel times drop significantly.

4. Reflect on which family (AR, ARMA, ARIMA) provides the most stable vs. the most reactive estimate of accident clearing.
   Based upon results in Figure 7, if we want fast reaction, then AR(1), if we want a smooth and reliable estimate, then ARMA, if we want a long-term trend but can tolerate lag, then ARIMA(d = 1).

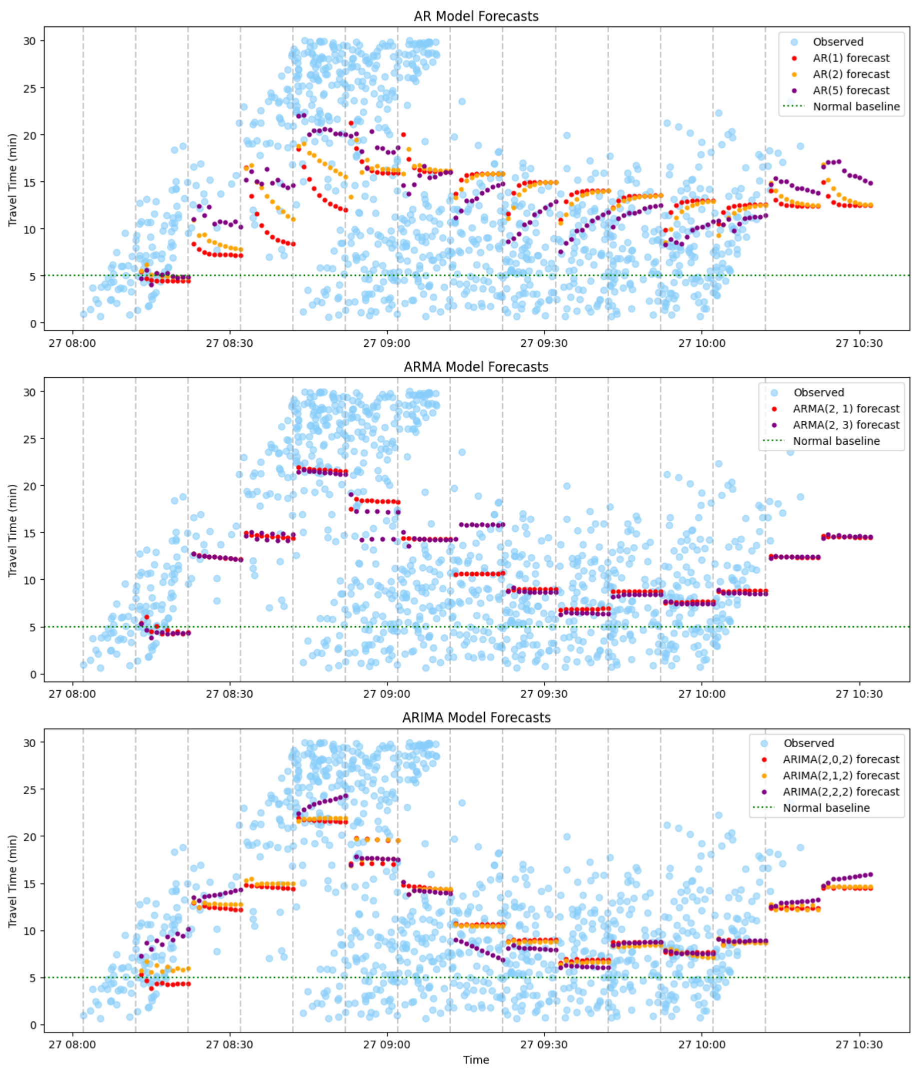


Figure 7: AR, ARMA, ARIMA model forecasts

3 Accident Start and Clearance (Changepoint Detection)

In Part 3, we see accidents as changepoints in the traffic data, as they cause a distribution shift in the data. We apply three changepoint detection techniques (CUSUM test, Page-Hinkley test, PELT test) described below.

3.1 CUSUM

Results of CUSUM, which is a basic change point detection technique, are described in the Appendix section.
Discussion Questions:

1. How sensitive detection is to the choice of h?
   When h is small, CUSUM triggers easily, meaning a sharp rise in durations causes S^+ to cross early. As h increases (10 or 15), the test becomes more conservative, requiring a longer accumulation of deviations before triggering a detection. This reduces false positives but also delays or sometimes misses the true accident start and clearance.

2. How results differ when using a fixed mean vs. a running average?
   For a fixed baseline mean, we keep the reference mean fix at a constant value. This makes S^+ rise quickly when travel times suddenly increase, which is good for detecting the start of an accident. While Running mean adapts to new traffic conditions, it detects both the start and the clearance more accurately.

3. Why tracking both upward and downward deviations can be useful?
   CUSUM uses S^+ to detect when travel times suddenly increase (slower traffic) and S^- to detect when travel times suddenly decrease (faster traffic). This is useful because an accident causes a mean increase in travel times, while the accident's clearance causes a mean decrease back toward normal. If we only tracked one direction, we could detect either the start or the clearance. By maintaining both S^+ and S^-, the method can detect both giving full changepoint detection.

3.2 Implementing Page–Hinkley

Results of the Page-Hinkley Test, a change point detection technique, are described in the Appendix section.
Discussion Questions:

1. How does the tolerance parameter (δ) help reduce false alarms compared to plain CUSUM?
   The tolerance δ ignores small deviations from the mean. When δ is small, every small wiggle in the data creates a changepoint, showing lots of false

alarms. When δ is larger, the detector stops reacting to tiny noise and only flags sustained changes, reducing false positives compared to CUSUM. In my results below, $\delta = 30$ gives robust result for change-point detection, whereas when $\delta = 20$, it detects many change-points that might be the effect of random noise.

2. What trade-off do you observe when increasing the threshold? (Think: sensitivity vs. missed detections.)
   A higher threshold makes the detector less sensitive, so we get fewer detections.

3. Why might the forgetting factor (α) be useful in non-stationary traffic data?
   Traffic is not a stationary series; it naturally changes, i.e., over the morning (rush hour, clearing, mid-day slowdown). α lets the running baseline mean adjust slowly to these trends. Without α: Morning traffic increases would look like a fake "accident" to the system, while with α: The detector adapts to gradual changes and only flags unexpected jumps, not normal daily trends.

4. What are the advantages and disadvantages of using a package over writing your own code from scratch, for these simpler statistical tests?
   Using a package saves time, reduces the chance of bugs, and provides a reliable, well-tested implementation. But, it also limits how much we can customise the internal mechanics of the algorithm.

3.3 Implementing PELT

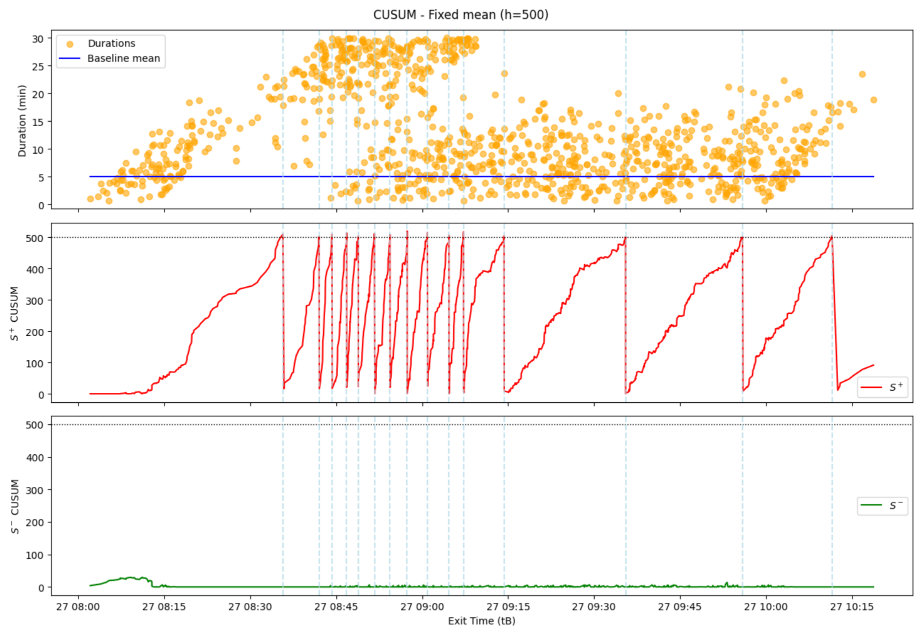
Results of the PELT test on our traffic data are attached in the Appendix section.
Discussion Questions:

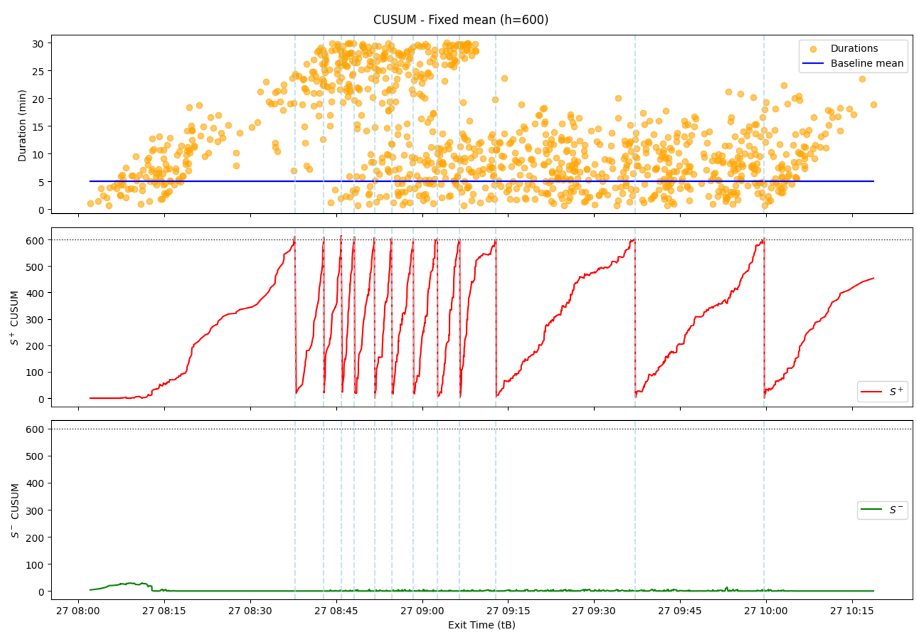
1. How does the choice of penalty affect the number of detected change-points?
   A smaller penalty detects more of the changepoints, sometimes even over-fitting to noise. A larger penalty makes changepoints more expensive, so only sustained shifts in the mean are detected. In my results, when the penalty increased from 2 to 20, the detected changepoints varies according to the way mentioned above.

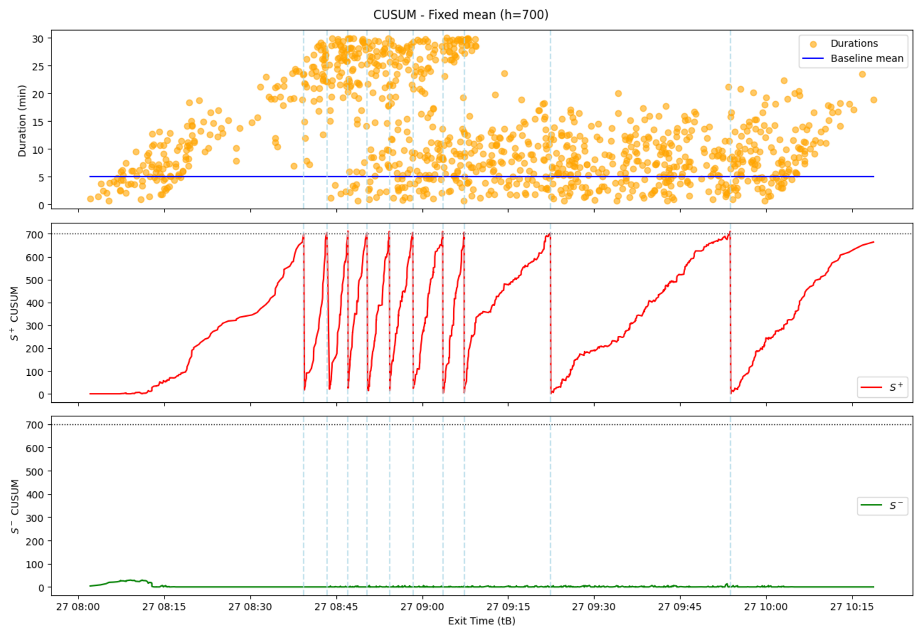
2. What are the advantages of an offline method like PELT compared to online methods such as Page–Hinkley?
   PELT has the advantage of analysing the entire sequence at once, which allows it to revisit earlier decisions and choose the segmentation that best explains the whole dataset globally. This makes it more stable than online detectors, which must decide in real time and cannot correct earlier errors.

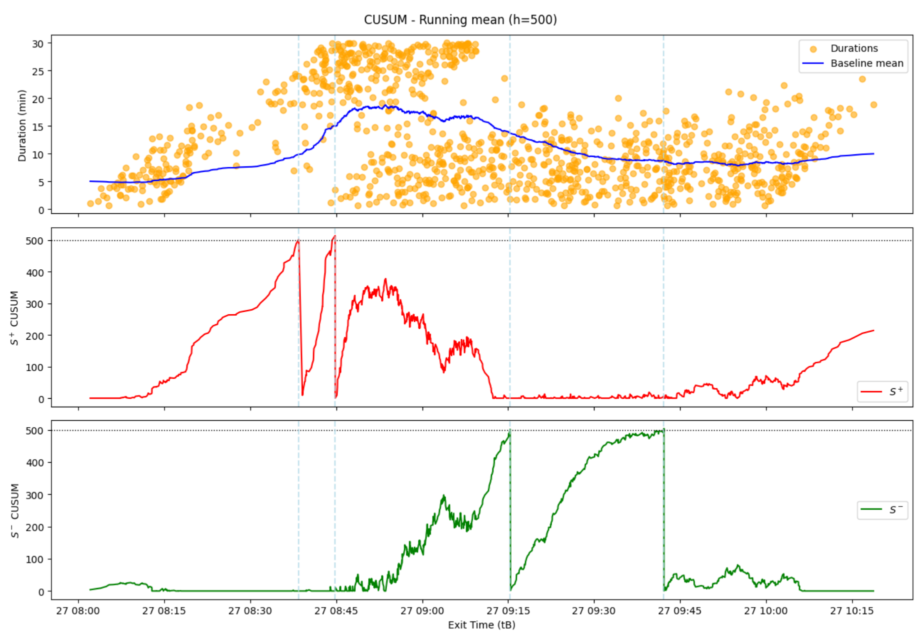
3. In what situations would you prefer an online detector over an offline one? And vice versa?
   Online detectors such as CUSUM or Page–Hinkley are preferred when the system must react immediately, for example, in live accident detection, fraud detection, where waiting for the full dataset is difficult. Offline detectors like PELT are more suitable when the full time series is available after the event, and the goal is accurate analysis.

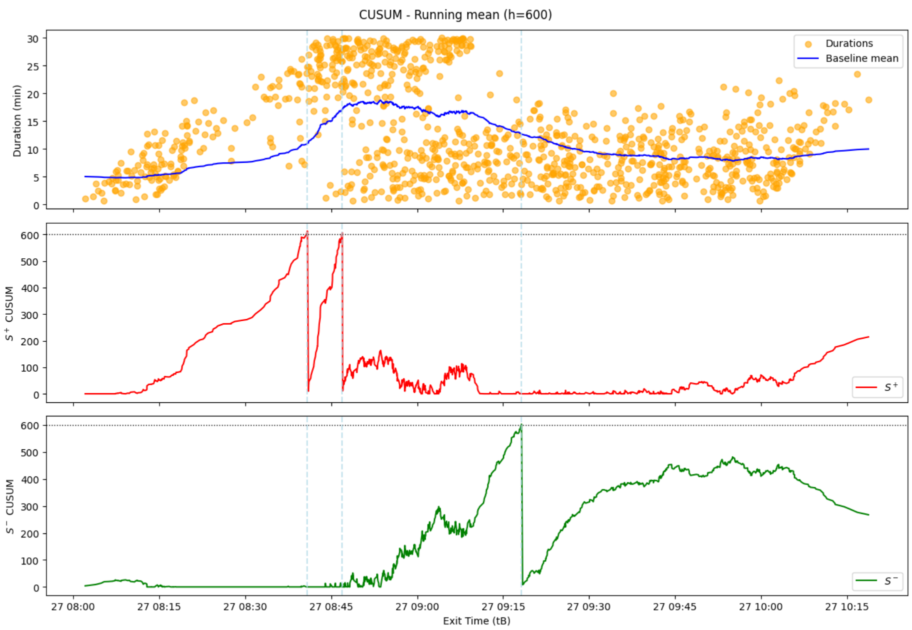
3.4 Appendix


CUSUM - Fixed mean (h=500)
Durations
Baseline mean
Duration (min)
S^+ CUSUM
S^- CUSUM
S^+
S^-
Exit Time (tB)
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15

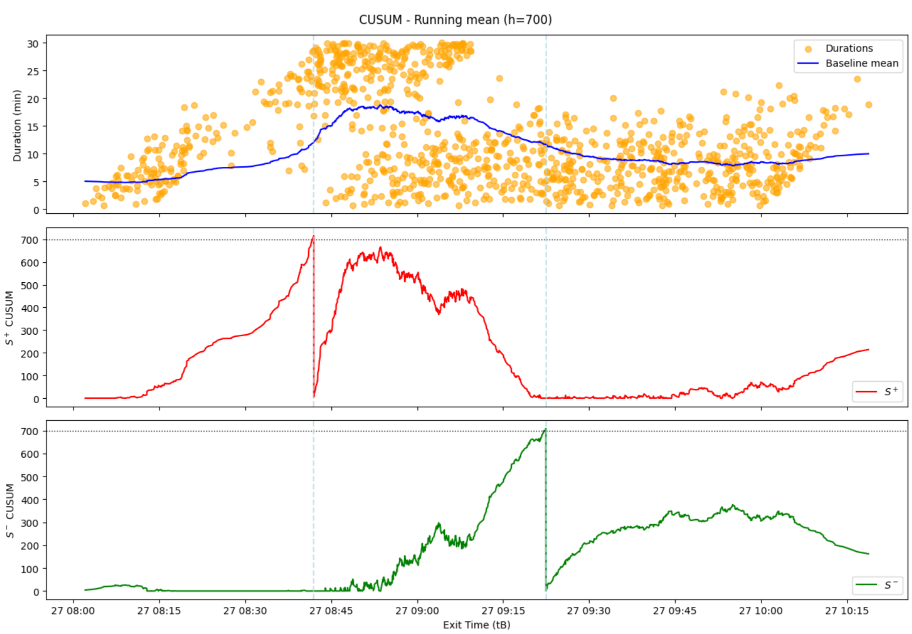

CUSUM - Fixed mean (h=600)
Durations
Baseline mean
Duration (min)
S^+ CUSUM
S^- CUSUM
S^+
S^-
Exit Time (tB)
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15


CUSUM - Fixed mean (h=700)
Duration (min)
Durations
Baseline mean
S^+ CUSUM
S^+
S^- CUSUM
S^-
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15
Exit Time (tB)

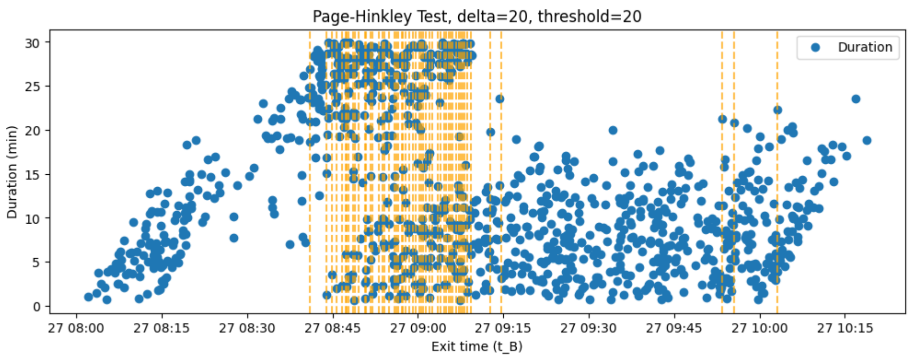

CUSUM - Running mean (h=500)
Duration (min)
Durations
Baseline mean
S^+ CUSUM
S^+
S^- CUSUM
S^-
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15
Exit Time (tB)

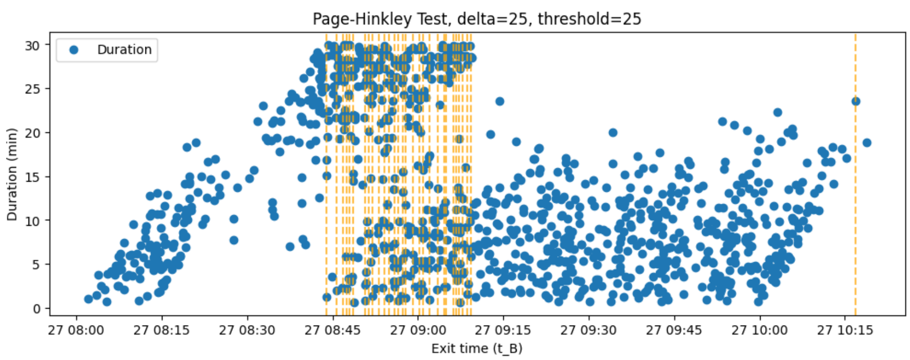


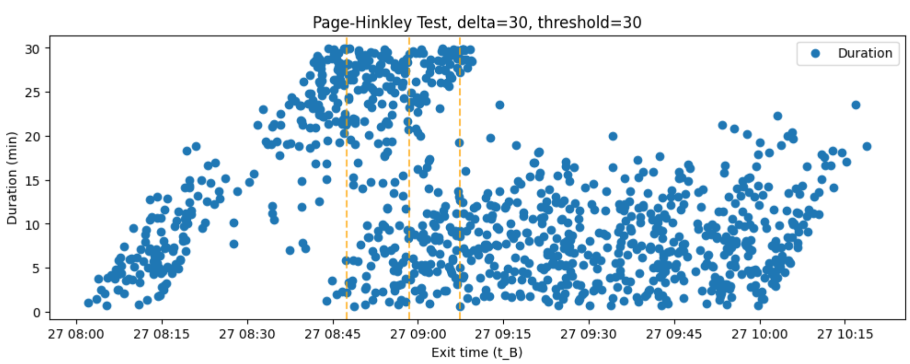
CUSUM - Running mean (h=600)
Durations
Baseline mean
Duration (min)
S^+ CUSUM
S^- CUSUM
S^+
S^-
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15
Exit Time (tB)

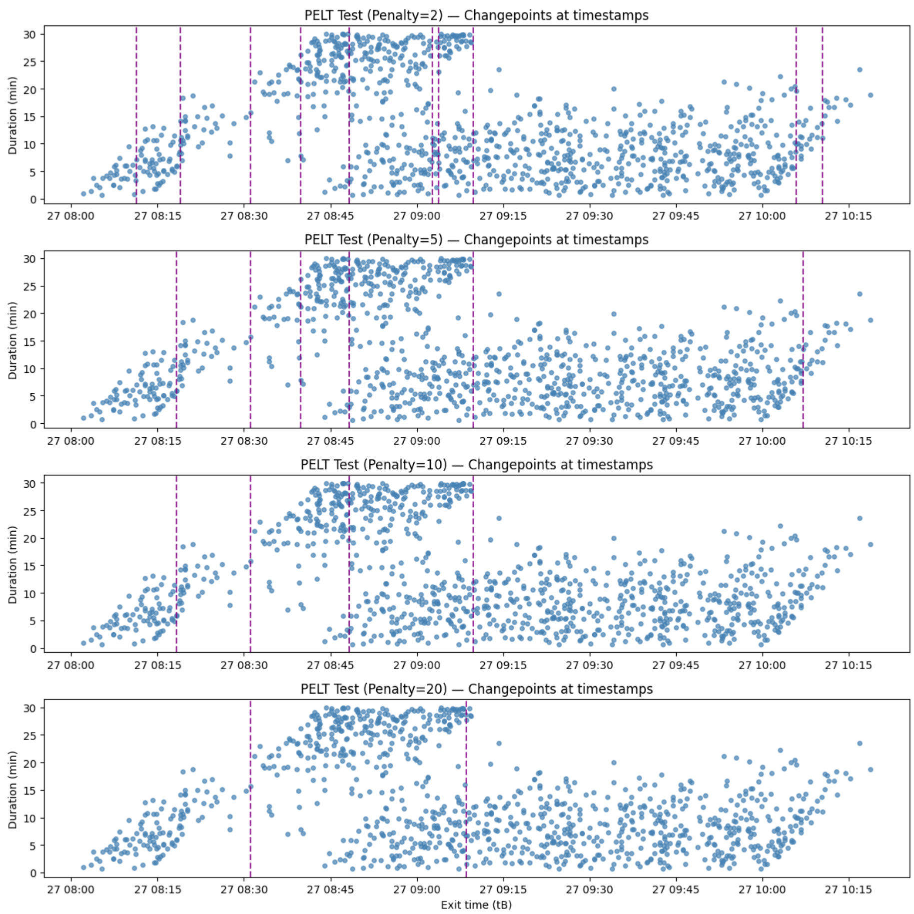


CUSUM - Running mean (h=700)
Durations
Baseline mean
Duration (min)
S^+ CUSUM
S^- CUSUM
S^+
S^-
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15
Exit Time (tB)


Page-Hinkley Test, delta=20, threshold=20
Duration
Duration (min)
Exit time (t_B)
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15


Page-Hinkley Test, delta=25, threshold=25
Duration
Duration (min)
Exit time (t_B)
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15


Page-Hinkley Test, delta=30, threshold=30
Duration
Duration (min)
Exit time (t_B)
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15


PELT Test (Penalty=2) — Changepoints at timestamps
PELT Test (Penalty=5) — Changepoints at timestamps
PELT Test (Penalty=10) — Changepoints at timestamps
PELT Test (Penalty=20) — Changepoints at timestamps
Duration (min)
0
5
10
15
20
25
30
27 08:00
27 08:15
27 08:30
27 08:45
27 09:00
27 09:15
27 09:30
27 09:45
27 10:00
27 10:15
Exit time (tB)