Official CUSUM Report

Gavin Croft

November 2025

Contents

1 Overview

This report summarizes the version of **CUSUM (cumulative sum)** used in the Intraday-Volatility-Monitor (IVM) project. CUSUM detects small standard deviation, persistent shifts in a process mean over time.

For our purposes, we will use CUSUM to look at the standard deviation, this will allow us to detect changes in movement without requiring a stable mean. This makes the approach more suitable for minute-by-minute trading environments.

2 The Use of CUSUM In The IVM

In the IVM project, CUSUM is applied to detect persistent directional movement in intraday price changes. Rather than computing deviations from a fixed expected mean, we evaluate each new price update relative to the previous minute.

Let

$$r = r_t - r_{t-1}$$

represent the most recent return difference, and let ε represent a volatility-scaled tolerance based on recent standard deviation estimates.

We then compute:

$$S_+ = \max\{0, r + \varepsilon\}$$

$$S_- = \max\{0, -(r + \varepsilon)\}$$

These act as accumulating evidence measures for upward and downward movement (these are sets of numbers that will keep growing in size). A detection event (alarm) occurs when the sum of these sets exceed our choice of threshold, h:

$$S_+ + S_- \geq h \Rightarrow \text{alarm}$$

and if

$$S_+ + S_- < h \Rightarrow \text{no alarm}$$

This allows our system to detect consistent directional movement that exceeds what we want to call 'normal volatility', even when the mean price level is drifting intraday.

3 Required Inputs

To run the intraday CUSUM algorithm, the following inputs are required:

- 1-minute trade data (price and timestamp)
- recent return difference r
- standard deviation estimate ε (volatility scaling factor)
- threshold h controlling sensitivity
- rolling 1-minute sampling

Unlike traditional CUSUM, no global mean μ_0 is required—the method adapts purely to relative movement (standard deviation).

4 Expected Outputs

The IVM CUSUM system will produce:

- positive and negative cumulative signals S_+, S_-
- detection events when $S_+ + S_- \geq h$

- timestamps and magnitude of detected moves
- direction of movement (positive or negative). These will be shown as charts of the changes in positive and negative volatility.

These outputs support real-time monitoring of intraday volatility shifts and trading opportunities.

5 Implementation (Pseudo-Code)

Below is a frame work of how the team will implement the initial steps of the CUSUM algorithm. This pusedo code (threshold value and reset condition) isn't currently optimized, but will become more fine tuned as the team runs data through the algorithm and we observe alarm rates.

```
def CUSUM (start, end, k, h):
    Out = []        //used for storing output
    lastReset = 0 //keeps track of time since last reset
    S_plus = 0
    S_minus = 0
    for (i = start to end):
        x = fetchData(i)    //get the ith data point
        S_plus += max(0, x + k)
        S_minus += max(0, -(x+k))
        S = S_plus + S_minus
        Out.append(S)   //add data point to Out
        lastReset += 1
        if (S > h):
            print("alarm: increased volatility")

        //reset sums when S hasn't moved recently
        if (lastReset > 10):
            pctChange = (S/Out[i-start-10]) -1  //pct change over last 10 min
            if (pctChange < 0.1):
                S_plus = 0
                S_minus = 0
                lastReset  = 0

    return Out
```

6 Issues and Limitations

When applying this CUSUM approach to high-frequency intraday data, the following issues may arise:

1. Non-stationary: volatility and price structure evolve throughout the day.
2. Noise sensitivity: rapid micro-price movements may trigger partial accumulations.
3. Threshold selection: h must balance false alarms vs. slow detection.
4. Lag: evidence must accumulate before triggering.
5. Computation: rolling minute-by-minute updates must remain efficient.

7 Conclusion

CUSUM provides an effective real-time detection mechanism for intraday directional volatility patterns. By basing the calculation on standard deviation, the implementation aligns with the dynamic nature of high-frequency trading.

Once we tune it correctly tuned correctly (by selecting an appropriate h), the method can highlight meaningful movements while filtering out normal market noise, strengthening the IVM team's ability to detect emerging intraday anomalies as soon as possible.