Student: Ilkin Mirzayev

Project Report: Detecting Anomalies in Credit Card Transactions

1. Understanding the CUSUM Principle

This project focused on using the **CUSUM (Cumulative Sum) algorithm**, which is a clever statistical method, to detect suspicious credit card transactions in real-time.

Unlike simpler monitoring tools that only look at one transaction at a time, CUSUM is powerful because it **keeps a running total** of how much transactions deviate from the normal average, or baseline (μ_0). This makes it really good at spotting small, consistent shifts that might be missed otherwise.

Essentially, CUSUM answers the question: "Has the typical transaction behavior changed enough to worry about?"

The Core Idea

CUSUM tracks two scores as each new transaction (x_t) arrives:

1. **Positive CUSUM (**S_t^+**):** This score goes up when transactions are consistently *too high* (looking for sudden spending increases).
2. **Negative CUSUM (**S_t^-**):** This score goes up when transactions are consistently *too low* (looking for sudden spending decreases).

The score only increases if the transaction is outside a small margin defined by the **Tolerance Constant (**k**)**. If either score passes the **Decision Threshold (**h**)**, we get an alert. After an alert, the score is reset to zero, and we start tracking again.

2. The Data We Used

We ran this analysis on a file called transactions_100k_clean.csv, which contains 100,000 simulated credit card transactions.

- **Normal Transactions:** Most transactions naturally hover around the expected average, which we set as $\mu_0 = \$50.0$.
- **Anomalous Transactions:** The data includes some transactions that were deliberately designed to be outliers—either much higher or much lower than normal—to simulate fraudulent activity.

3. Implementation and Key Findings

Our Python program cusum_detector.py worked like a stream-processing tool, analyzing the data one transaction after another.

CUSUM Settings:

Parameter	Value	What it Means
Reference Mean (μ_0**)**	50.0	The expected average transaction in dollars.
Tolerance Constant (k**)**	5.0	We ignore deviations of less than k per transaction.
Decision Threshold (h**)**	20.0	The total accumulated deviation needed to trigger an alarm.

Summary of Results

Metric	Result
Total Transactions Examined	100,000
Total Anomalies Spotted by CUSUM	**6,151**
Average Absolute Deviation of Anomalies	**$40.71**

With our current settings, the system flagged 6,151 anomalies. Since the average deviation is over $40, it confirms that the CUSUM successfully flagged transactions that were significantly different from the baseline of $\$50$.

A Few Examples of Anomalies Found:

ID	Timestamp	Amount	Deviation	S_t^+ / S_t^-	Type
55	2025-01-01T00:00:54	296.98	246.98	241.98 (S_t^+)	Positive (High)
64	2025-01-01T00:01:03	26.81	-23.19	32.64 (S_t^-)	Negative (Low)

4. Discussion on How to Tune the CUSUM Detector

The performance of a CUSUM system depends heavily on how we set k and h. This is crucial for balancing speed and accuracy in a real system.

A. Tuning the Tolerance Constant (k)

k dictates how much small variations are allowed before they even start to add up.

- **If you use a high** k **(e.g.,** $k=15$**):** The system becomes less sensitive. It's great for reducing **False Alarms (FAR)**, but it will take longer to catch a genuine problem.
- **If you use a low** k **(e.g.,** $k=1$**):** The system becomes hyper-sensitive, adding up even tiny deviations quickly. You'll catch issues faster (lower **Average Run Length, or ARL**), but you'll get many more alerts for normal process noise.

B. Tuning the Decision Threshold (h)

h is the final alarm limit—how high the total cumulative score can go before we raise an alert.

- **If you use a high** h **(e.g.,** $h=50$**):** The detector is very cautious. Only really large, persistent shifts will trigger an alarm. This is useful if you only care about critical changes.
- **If you use a low** h **(e.g.,** $h=5$**):** The detector is much more aggressive. Even a short burst of suspicious activity will set it off. This is preferred when the cost of missing fraud is extremely high.

5. Summary

The CUSUM algorithm is a really effective solution for monitoring continuous data streams like financial transactions. It's efficient because it only requires keeping track of two numbers (S^+ and S^-), saving memory. Most importantly, it's designed to catch the subtle, gradual fraud schemes that would easily bypass simpler detection methods. By carefully tuning k and h, financial institutions can fine-tune the system to meet their specific security needs.