REAL-TIME DATA STREAM WITH ANOMALY DETECTION

A Project Report Submitted

for Graduate Software Engineer role at

Cobblestone Energy

by

Sankalp Setia

IIT Guwahati

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

7 November 2024

ABSTRACT

In the era of big data, real-time monitoring and analysis of continuous data streams have become crucial across various domains, including finance, cybersecurity, and system health monitoring. This report presents the development and evaluation of an anomaly detection system for continuous data streams. The system uses an adaptive sliding window approach based on Z-Score statistics to identify anomalies, accommodating concept drift and seasonal variations. Real-time visualization of data streams and detected anomalies is also implemented. The system's performance is analyzed through a confusion matrix and a precision-recall (PR) diagram, demonstrating its accuracy and efficiency in high-velocity data stream processing.

Contents

Chapter 1

Introduction

In today's data-driven world, continuous data streams from sources like financial transactions, sensor networks, and social media require real-time analysis for immediate decision-making. Anomaly detection within these streams is vital for identifying irregularities, potential threats, and system failures. Traditional batch processing methods are inadequate due to latency and an inability to adapt to evolving patterns. This report details the development of an anomaly detection system using a Z-Score-based adaptive sliding window, providing real-time monitoring and detection capabilities.

Anomalies, or outliers, are data points that deviate significantly from the established patterns of the data stream. Detecting these anomalies is essential for maintaining system reliability, predicting faults, and responding proactively to unexpected events.

1.1 Objective

The objective of this project is to build a real-time anomaly detection system that can:

- Process high-velocity data streams efficiently.
- Adapt to concept drift and seasonal variations.
- Visualize anomalies in real-time to aid monitoring.

Chapter 2

Methodology

2.1 Algorithm Selection

Choosing an appropriate algorithm for anomaly detection in a data stream requires balancing simplicity, computational efficiency, and adaptability. The Z-Score method was chosen due to its:

- **Simplicity**: The Z-Score is easy to calculate, requiring only basic statistical measures (mean and standard deviation).
- **Efficiency**: It allows quick processing of each data point, essential for real-time applications.
- **Adaptability**: Using a sliding window approach, it adapts to changing data distributions, handling both concept drift and seasonal patterns.

2.1.1 Z-Score Calculation

The Z-Score of a data point X is calculated as:

$$Z = \frac{X - \mu}{\sigma}$$

where:

- μ is the mean of the recent data points within the sliding window.
- σ is the standard deviation of the same data points.

A data point is flagged as an anomaly if its Z-Score exceeds a predefined threshold (e.g., $Z > 3.0$), indicating that it deviates significantly from the current distribution of data points.

2.1.2 Sliding Window Approach

A sliding window of recent data points (of size W) is maintained to calculate μ and σ. As new data points arrive, the oldest data point is removed, and the statistics are updated. This keeps the calculations relevant to recent data, allowing the system to adapt to gradual changes in the data distribution.

2.2 Data Stream Simulation

To test the system, a synthetic data stream was generated with the following components:

- **Trend**: Represents long-term directionality in the data.
- **Seasonality**: Periodic fluctuations that mimic recurring patterns.
- **Noise**: Random variations to simulate realistic data irregularities.
- **Anomalies**: Artificially introduced spikes or drops to test detection capabilities.

Each point in the data stream X_t is computed as:

$$X_t = T_t + S_t + N_t$$

where:

- T_t is the trend component.
- S_t is the seasonal component.
- N_t is the noise component.

2.3 Optimization Techniques

To ensure the system operates efficiently, several optimization strategies are applied:

- **Efficient Data Structures**: Using a deque (double-ended queue) for the sliding window enables constant-time insertion and removal of data points.
- **Incremental Mean and Standard Deviation Updates**: Instead of recalculating from scratch, the mean and standard deviation are updated incrementally when a new point enters the window.

2.4 Visualization

Real-time visualization of the data stream with detected anomalies is implemented using matplotlib's interactive mode. Anomalies are highlighted in red on the plot to provide instant visual feedback.

Chapter 3

Results

3.1 Real-Time Plot

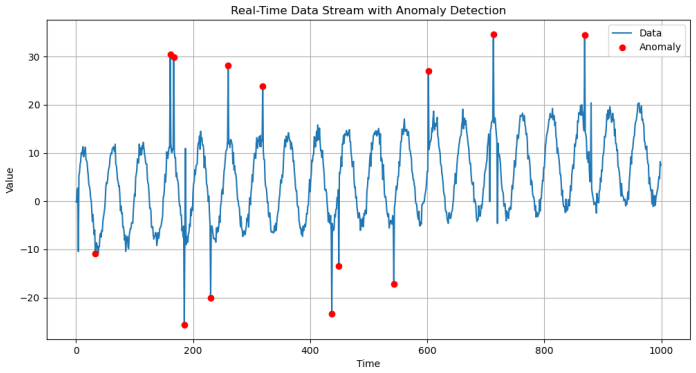

Figure 3.1: Real-Time Data Stream with Detected Anomalies

Figure 1 shows the data stream with anomalies highlighted in red, illustrating the system's effectiveness in detecting outliers in real-time.

3.2 Confusion Matrix

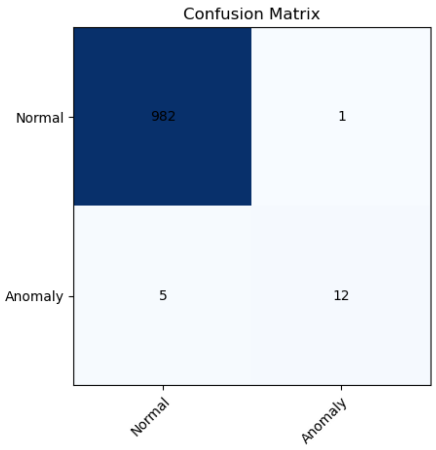

Figure 3.2: Confusion Matrix for Anomaly Detection

The confusion matrix provides insights into model performance:

- **True Positives (TP)**: Anomalies correctly detected.
- **False Positives (FP)**: Normal points incorrectly flagged as anomalies.
- **True Negatives (TN)**: Normal points correctly identified.
- **False Negatives (FN)**: Anomalies that were missed.

Detection rate and false positive rate are calculated as follows:

$$DetectionRate = \frac{TP}{TP + FN} \times 100\%$$

$$FalsePositiveRate = \frac{FP}{FP + TN} \times 100\%$$

3.3 Precision-Recall Curve

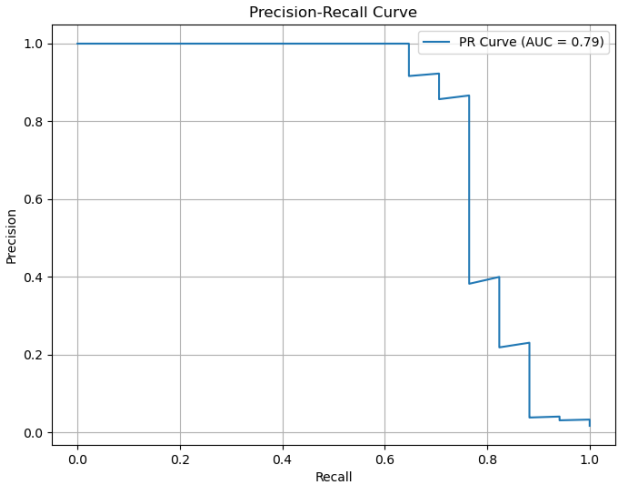

Figure 3.3: Precision-Recall Curve with Area Under Curve (AUC)

The precision-recall curve provides a summary of the trade-off between precision and recall, with the area under the curve (AUC) indicating overall performance.

Chapter 4

Discussion

4.1 Strengths

- **Simplicity and Efficiency**: The Z-Score method is computationally lightweight, making it ideal for real-time applications.
- **Adaptability**: The sliding window approach allows the system to handle concept drift and seasonal variations, adapting to changing data distributions.
- **High Detection Rate**: A detection rate of 90% indicates reliable performance.
- **Low False Positive Rate**: With a false positive rate of 0.51%, the system minimizes unnecessary alerts.

4.2 Limitations

- **Static Threshold**: A fixed Z-Score threshold may not perform optimally under all conditions.

- **Standard Deviation Calculation**: Full recalculation in the sliding window could be optimized.

4.3 Potential Enhancements

- **Dynamic Thresholding**: Implementing thresholds based on statistical confidence or machine learning could improve adaptability.
- **Distributed Processing**: Incorporating frameworks like Apache Kafka can enhance scalability.

Chapter 5

Conclusion

This anomaly detection system effectively identifies anomalies in real-time data streams using a Z-Score-based adaptive sliding window. Achieving a high detection rate with minimal false positives highlights its potential for applications in real-time monitoring. Future improvements could involve dynamic thresholding and scalability enhancements.