Implementation and Evaluation of PASAD on SWaT and TE Datasets

Vraj Patel - 24110080

Pavani Priya - 220415

Divya Krupa - 210274

Akshat Shrivastav - 220103

October 13, 2024

Contents

1 Introduction

1.1 Overview of PASAD

PASAD is a tool for detecting cyberattacks in industrial control systems by monitoring sensor data in real time. It identifies anomalies by spotting unusual changes in the data, which could signal an attack. PASAD uses mathematical techniques to find patterns in the data and flags anything that looks suspicious, without requiring a detailed model of normal operations. This makes it especially effective for detecting stealthy attacks.

1.2 Tasks Overview

The tasks performed in this report are as follows:

- Implement PASAD on the SWaT and TE datasets.
- Plot sensor measurements and departure scores.
- Compare two PASAD variants in terms of runtime and detection performance.
- Analyze five attack scenarios in the TE dataset and compare alarm counts.
- Explore the use of the centroid instead of the mean for the departure score.
- Implement Mahalanobis distance for the departure score and compare with previous methods.

2 Tasks

2.1 Task 1a: Dataset Analysis and Plotting

2.1.1 TE Dataset

For the TE dataset, training data is selected from measurements $N = 0$ to $N = 2000$, while the rest of the data is used for testing. This approach is consistent across all attack scenarios (SA1, SA2, SA3, DA1, and DA2).

Parameter Justification for TE Dataset: The reduced dimensionality parameter r is chosen based on the following conditions:

- $r > 1$: A condition specified in the problem.
- A plot of singular values against r: The value of r is selected just before a steep decline in the singular values to retain the components carrying significant information while discarding noise.

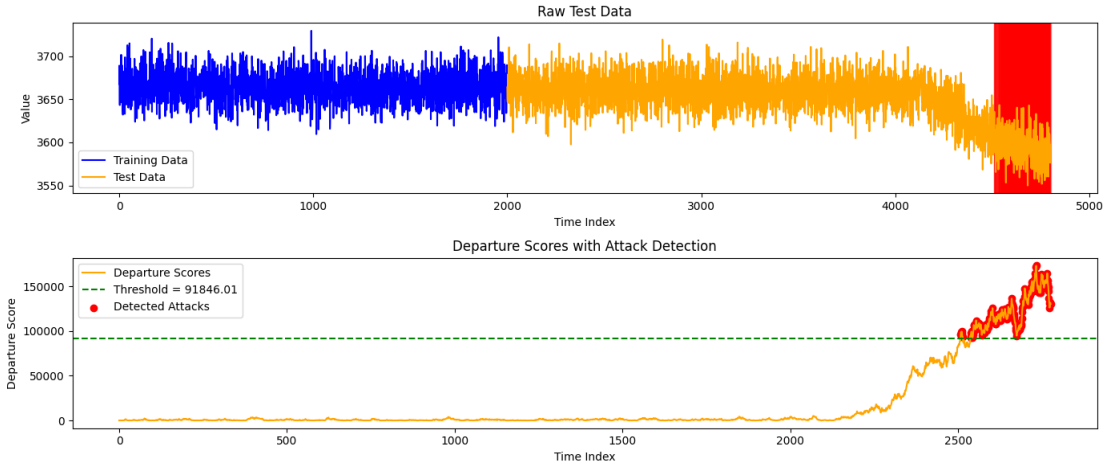
The lag parameter L is determined by analyzing the shape of the departure score plot. A low L results in erratic departure scores, while high L produces linear and scattered plots, both of which are unsuitable. An optimal value of $L = 24$ was selected as it balances pattern detection without introducing too much noise.

Results: The departure scores exceed the threshold, indicating anomalies in the data and successful attack detection (Figure 1).

2.1.2 SWaT Dataset

For the SWaT dataset, the normal dataset is used for training, while the attack dataset is used for testing as per the task requirements.

Figure 1: TE Dataset



Parameter Justification for SWaT Dataset: The reduced dimensionality parameter r is selected using the same approach as for the TE dataset, ensuring that noise is minimized while retaining significant components. Similarly, Lag parameter L is determined by analyzing the shape of the departure score plot. A higher Lag parameter could result in better detection but training memory requirements becomes too high!

Results: The departure scores exceed the threshold, indicating the detection of an attack in the SWaT dataset (Figure 2).

Figure 2: SWaT Dataset

2.2 Task 1b: Performance Comparison

Performance of PASAD Using UT vs. U^TU: PASAD's efficiency is improved by using the UT method instead of the traditional U^TU approach. The UT method leverages an isometry trick, avoiding explicit projection and reducing computational overhead.

Efficiency Gain:

- Detection time for UT: 0.00307 ms
- Detection time for U^TU: 0.00477 ms

The UT method is approximately 1.5 times faster than the U^TU method, with no loss of accuracy in anomaly detection (Figure 3).

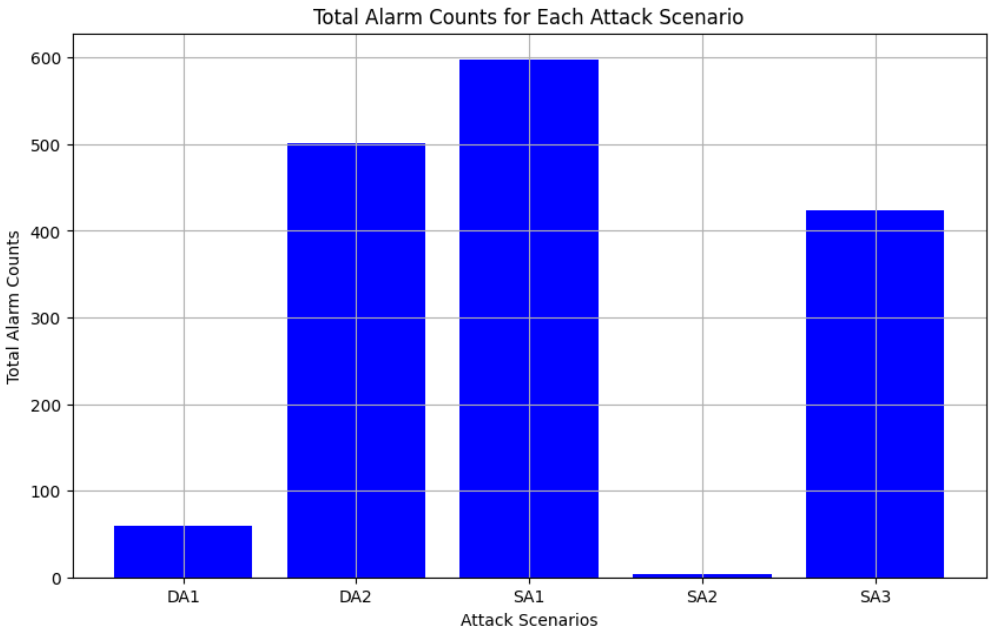
Figure 3: UT vs U^TU

2.3 Task 1c: Attack Scenario Analysis in TE Dataset

Methodology: To achieve a zero false alarm rate, the classifier threshold is set to the maximum departure score observed during the normal phase. Alarms are raised for all departure scores that exceed this threshold.

Results: (Figure 4)

Figure 4: Alerts for each dataset



2.4 Task 2: Exploring the Use of Centroid Instead of Mean for the Cluster

In this task, the departure score calculation is modified by using the midpoint of the data instead of the centroid. The midpoint is calculated as follows:

$$C = \frac{\min(X) + \max(X)}{2}$$

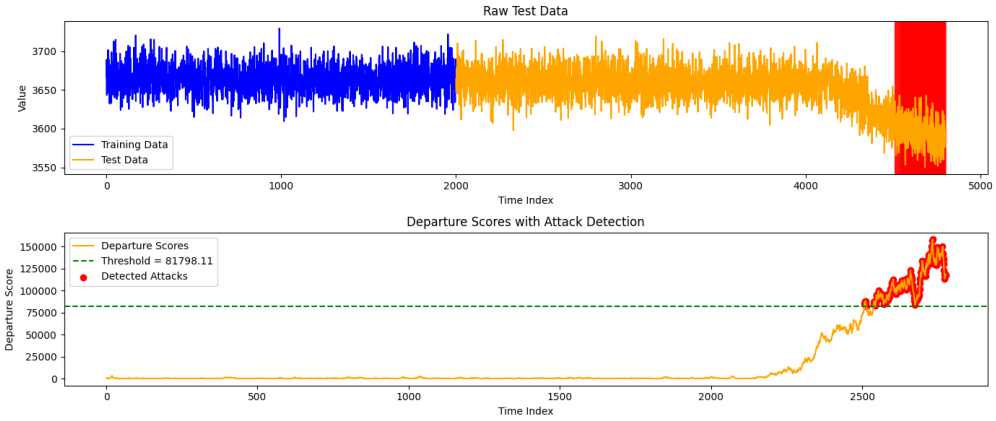
where X is the signal subspace matrix, and C is the midpoint.

Comparison:

- **Midpoint:** Considers only the extreme values (min and max), making it simpler but less effective.
- **Centroid:** Represents the average of all points, making it more accurate and less influenced by outliers.

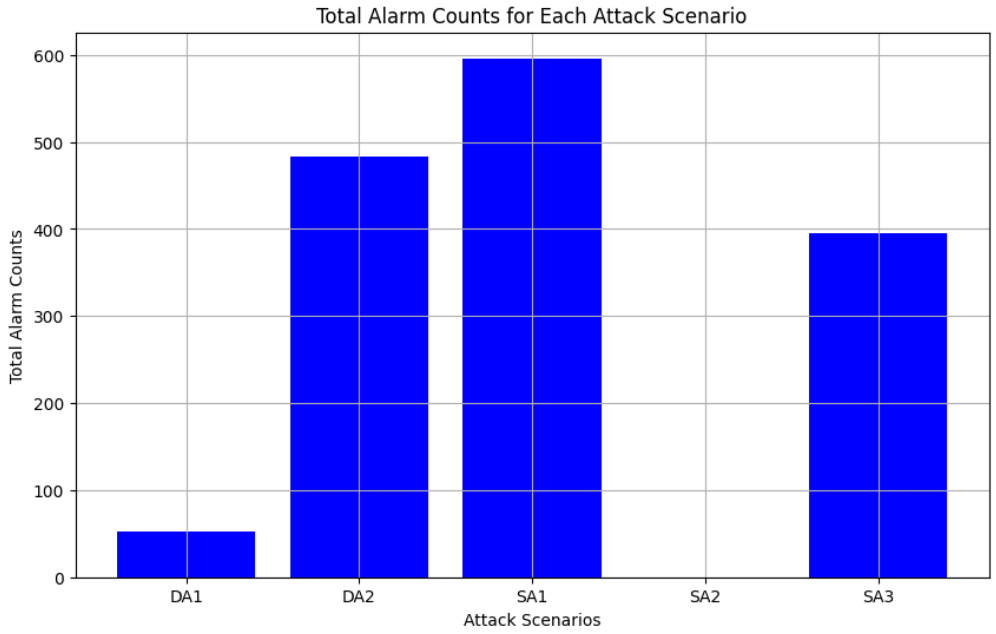
Effectiveness: The midpoint method is less effective than the centroid, as it doesn't capture the full structure of the data.

Figure 5: Alerts using midpoint on SA1



Barchart: Bar Graphs comparing Total Alert Counts Detected for Each Scenario using modified PASAD Method with midpoint.

Figure 6: Alerts for each dataset



2.5 Task 3: Exploring the Use of Mahalanobis Distance

In this task, Mahalanobis distance is used to calculate PASAD's departure scores and compared with the traditional Euclidean distance method.

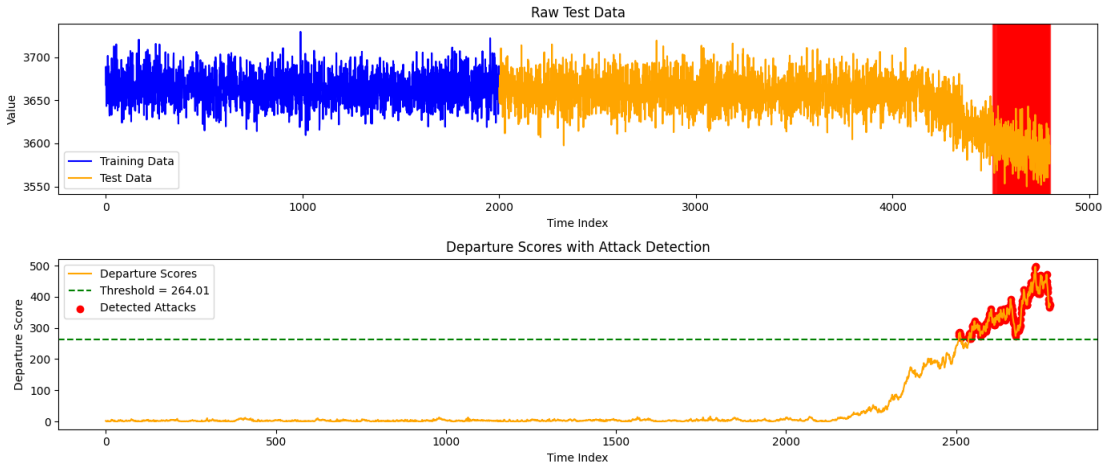
Euclidean vs. Mahalanobis Distance:

- **Euclidean Distance:** Measures straight-line distance, assuming all features are equally important.
- **Mahalanobis Distance:** Accounts for correlations between features, making it more accurate for complex data.

Impact: Mahalanobis distance provides improved detection accuracy when features are correlated or have different scales.

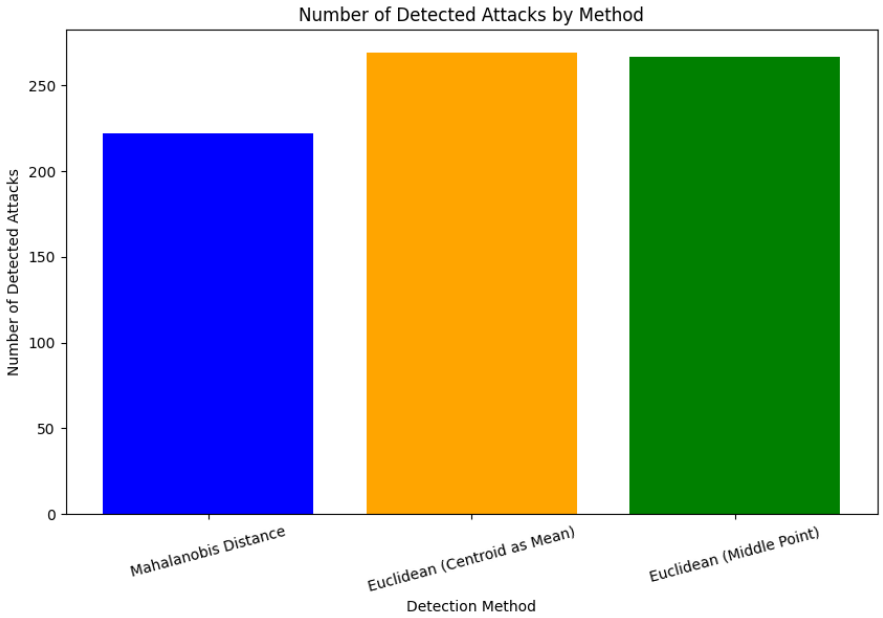
Results: Detection results using Mahalanobis Distance instead of Euclidean Distance.

Figure 7: Alerts using Mahalanobis Distnace on SA1



Comparison: Bar chart below shows number of Alter generated when Mahalanobis distance is used compared to when Euclidean Distance is used for calculations.

Figure 8: Comparing Alerts



Runtime Analysis: For the SA1 scenario:

- Runtime with Euclidean distance: 0.32s
- Runtime with Mahalanobis distance: 0.43s

While Mahalanobis distance offers better accuracy, it is computationally heavier.