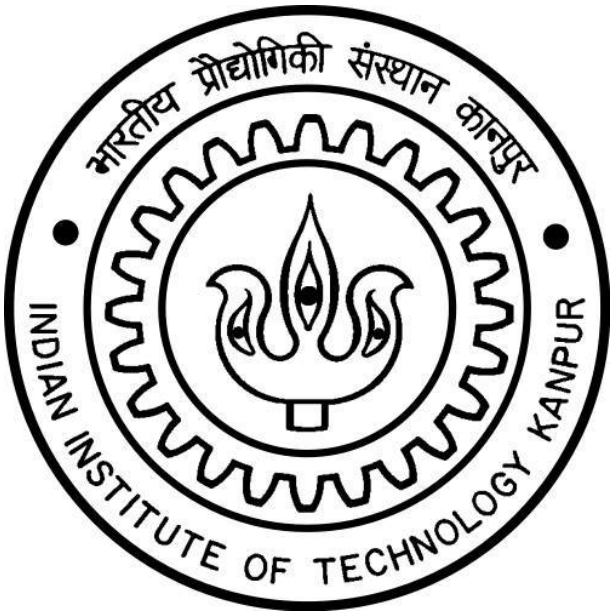


ASSIGNMENT ON "PROCESS AWARE STEALTHY ATTACK DETECTION(PASAD)"

Authors: Group 5

Himanshu Shekhar (220454)
Lokesh Yadav (220594)
Kamal Kant Tripathi (241110086)

October 13, 2024

1 Introduction

Process-Aware Stealthy Attack Detection (PASAD) is an advanced method used to detect cyber-physical attacks on industrial control systems (ICS). Unlike traditional security measures that focus on network traffic or software signatures, PASAD leverages the dynamics of physical processes themselves to identify anomalies indicative of a stealthy cyber attack.

Stealthy attacks, such as those targeting critical infrastructure (e.g., power grids, water treatment plants), are designed to remain undetected by conventional monitoring tools. Attackers manipulate the physical process slowly or in a manner that mimics normal operation, making it hard for typical detection systems to notice. PASAD, however, focuses on the subtle variations in the process signals like temperature, pressure, or electrical current using anomaly detection algorithms to spot deviations from expected behavior. By modeling the normal behavior of the physical system through data collected from sensors, PASAD establishes a baseline of acceptable performance. Any deviations from this baseline, especially those that do not trigger immediate alarms but could still degrade the system over time, are flagged as potential indicators of a stealthy attack.

PASAD's process-aware approach is particularly effective against zero-day attacks and insider threats, providing an extra layer of defense that can secure complex systems from malicious manipulation, while minimizing false positives.

2 Problem 1

The PASAD has been implemented by this Group No 5 using Python and the concepts provided in the class and paper on the subject. The **GitHub repository** containing the detail code can be found **here**.The implementation method has been depicted in succeeding subsections:

2.1 Embedding

In PASAD, Step 1 (Embedding) involves transforming a univariate time series $T = x_1, x_2, \cdots, x_N$ into a set of L-lagged vectors. Given a lag parameter L, this step embeds the time series into an L-dimensional space by constructing $K = N - L + 1$ lagged vectors, where each vector $x_i = (x_i, x_{i+1}, \cdots, x_{i+L-1})^T$ represents a sliding window of L consecutive measurements. These vectors are then organized into a trajectory matrix X, where each column corresponds to one lagged vector. This embedding step captures temporal dependencies and prepares the data for further analysis. The code can be referred from the link provided at section 2 above. We use $L = N/2$ in our implementation.

2.2 Singular Value Decomposition

In Step 2 (Singular Value Decomposition) of PASAD, the trajectory matrix X is decomposed using Singular Value Decomposition (SVD) to capture the essential structure of the time series. This process reduces noise and focuses on the core

deterministic behavior of the system. The SVD provides L eigenvectors $u_1, u_2, \cdots, u_L$ from the lag-covariance matrix XX^T. These eigenvectors represent the main directions of variability in the data. The number of significant eigenvectors, denoted as r, is chosen to capture the essential dynamics, effectively reducing the dimensionality for better anomaly detection. For choosing the value of r, we apply the criterion that the **consecutive difference in the sorted eigenValues should be maximum**.

NOTE: We apply this criterion ignoring the first eigenValue because it is typically disproportionately larger than the others, which makes it overshadow other eigenvalues. We get better detection results with $r > 1$ (i.e. not considering the first eigen value). Usually the first (leading) $2 - 20$ eigenvalues carry essential information about the signal, and this can only be visualized if we skip plotting the first eigenvalue.

2.3 Projection onto the Signal Subspace

In Step 3 (Projection onto the Signal Subspace) of PASAD, the goal is to mathematically represent the normal behavior of the process. Using the top r eigenvectors $u_1, u_2, \ldots, u_r$, an L-by-r matrix U is formed, which defines a subspace L_r representing the key dynamics of the system. The sample mean c of the lagged vectors is calculated, and its projection onto this subspace is computed as $\tilde{c} = Pc$, where $P = UU^T$ is the projection matrix. This centroid $\tilde{c}$ represents the central tendency of normal process behavior.

2.4 Distance Tracking

In Step 4 (Distance Tracking) of PASAD, the system monitors for attacks by calculating a departure score for each new observation. For each test vector x_j (with $j > K$), the squared Euclidean distance D_j between the vector's projection Px_j and the centroid $\tilde{c}$ in the signal subspace L_r is computed. This distance indicates how much the current behavior deviates from the learned normal process. If $D_j \geq \theta$ (a predefined threshold), an alarm is triggered, signaling a potential attack. This step allows for real-time detection of abnormal behavior in the system.

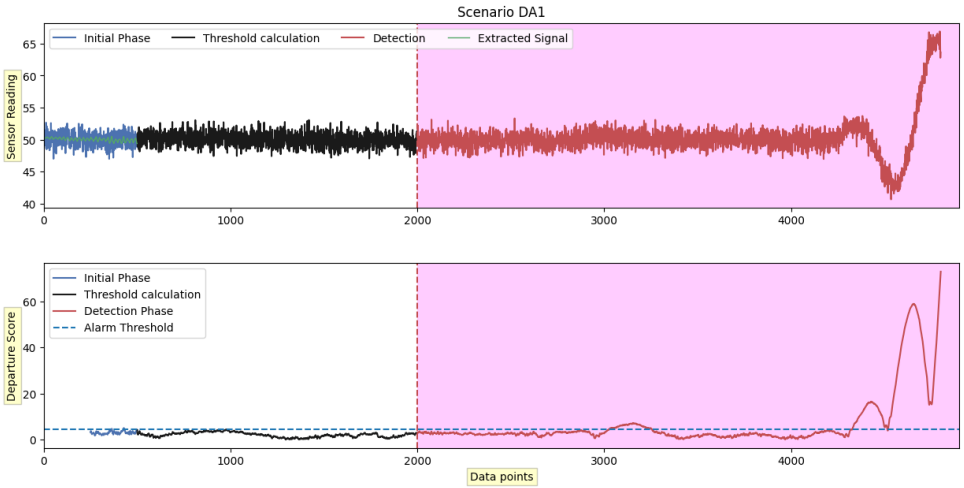
2.5 Results

2.5.1 Dataset Analysis and Plotting (TE Dataset)

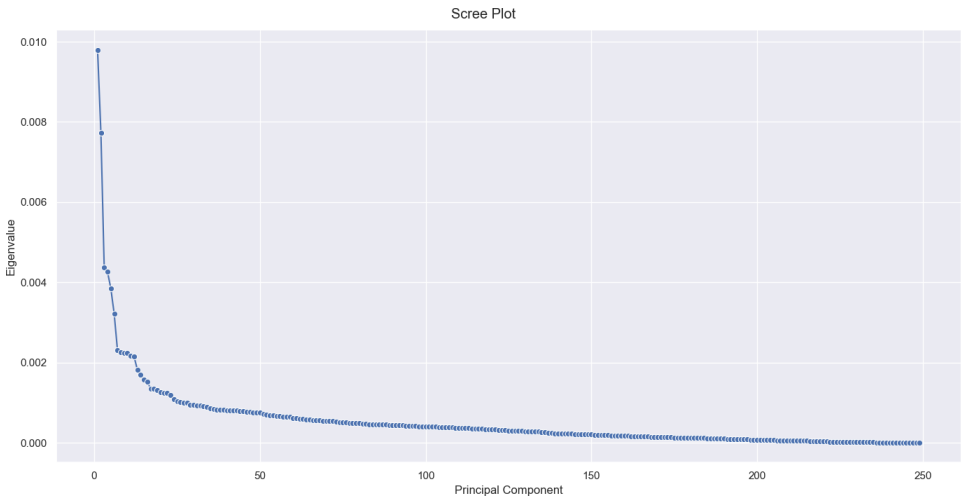
DA1 Scree Plot

Scenario DA1

DA1 - 14^{th} Column (0 - indexed)
Threshold value $= 4.501286770065579$
$N = 2000$
First 500 for initial phase and remaining for calculating threshold value.



DA2 Scree plot



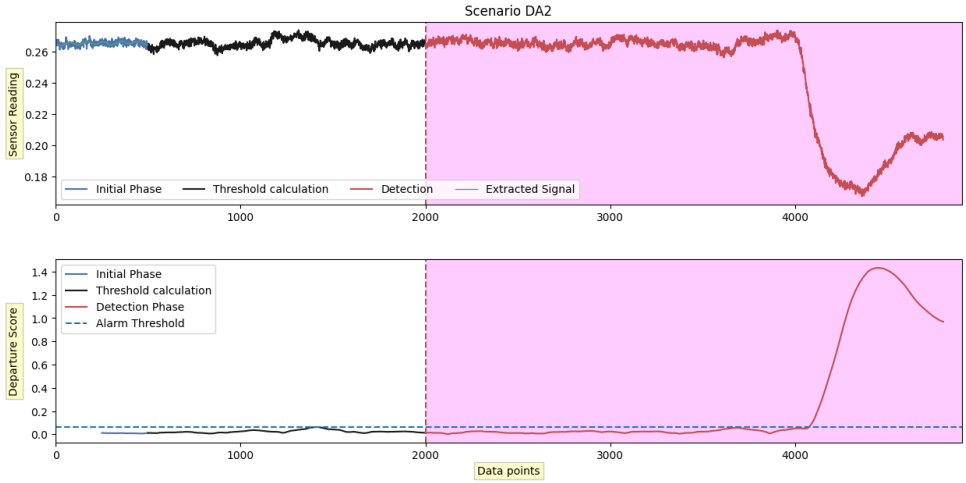
Scenario DA2

DA2 - 0^{th} Column

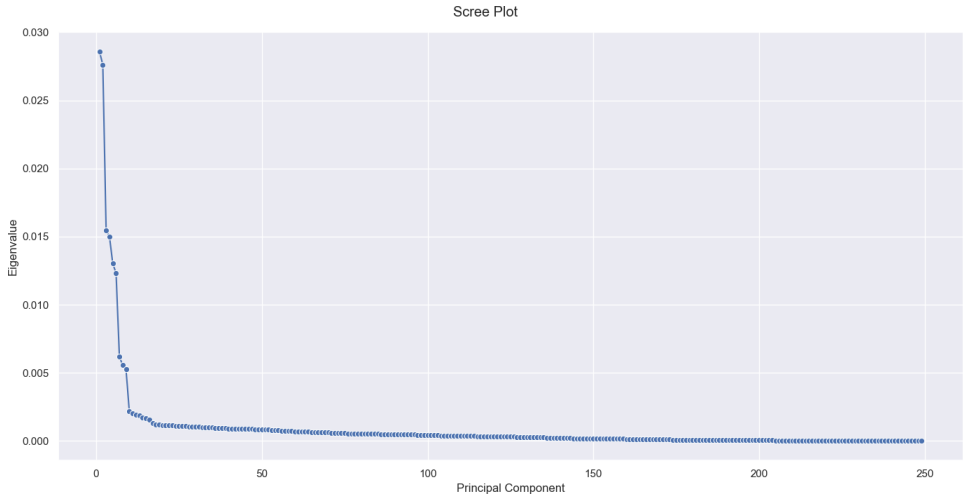
Threshold value $= 0.06242263896977938$

$N = 2000$

First 500 for initial phase and remaining for calculating threshold value.



SA1 Scree Plot



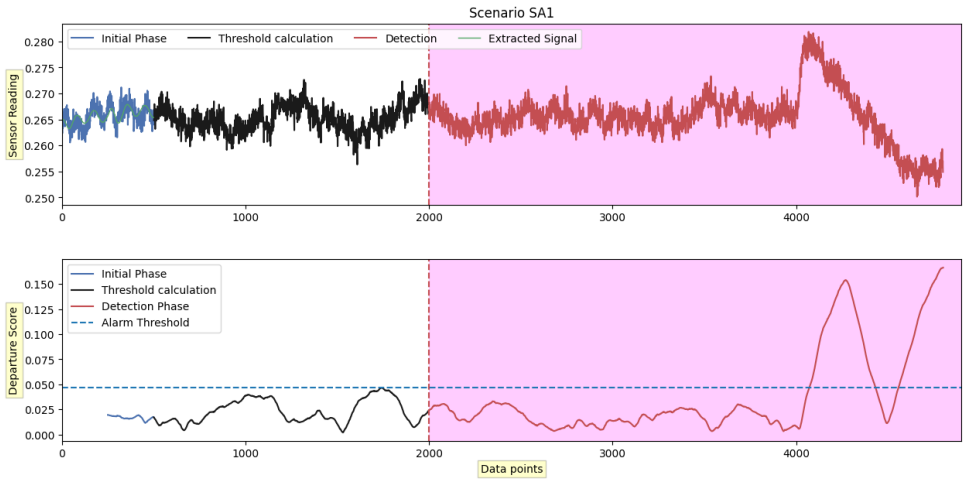
Scenario SA1

SA1- 0^{th} Column

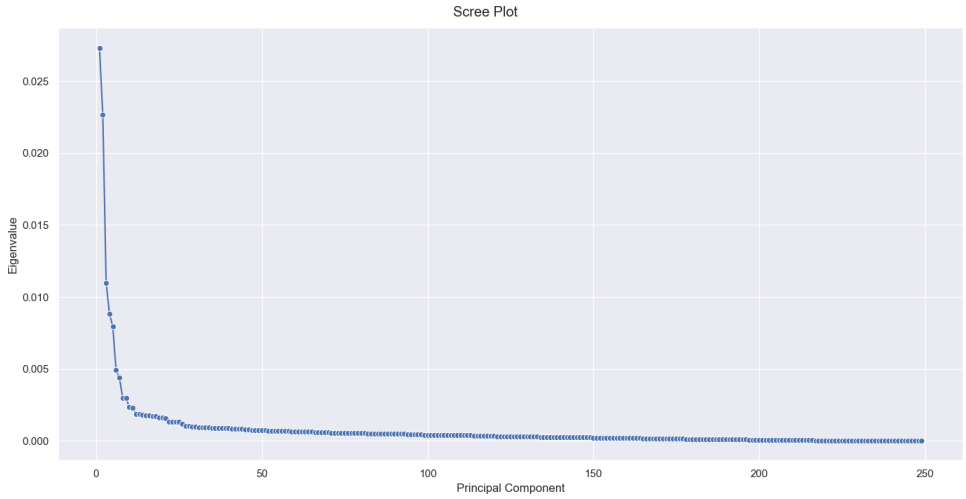
Threshold value $= 0.04684351083012187$

$N = 2000$

First 500 for initial phase and remaining for calculating threshold value.



SA2 Scree Plot



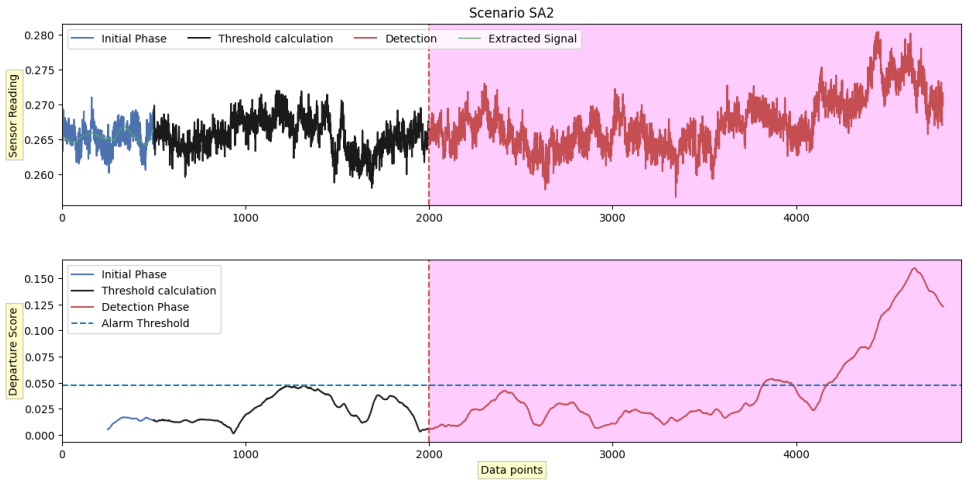
Scenario SA2

SA2- 0^{th} Column

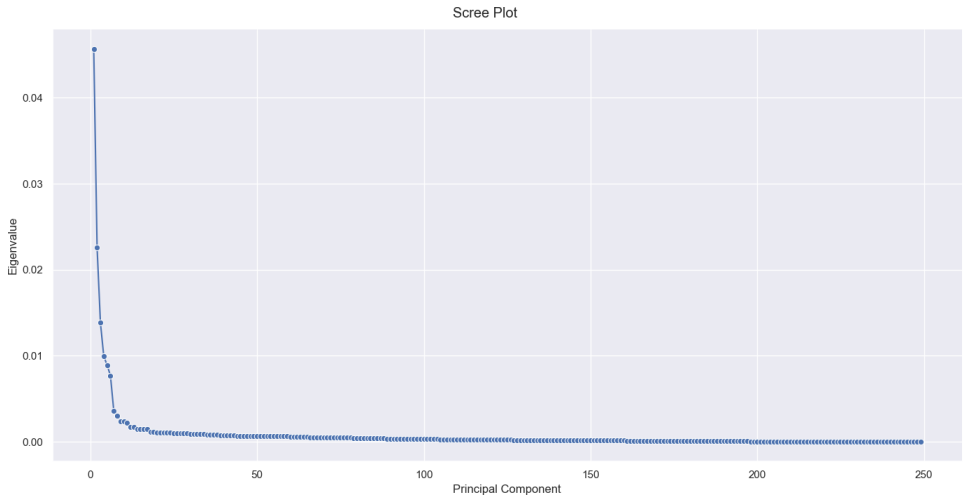
Threshold value $= 0.047040940181675615$

$N = 2000$

First 500 for initial phase and remaining for calculating threshold value.



SA3 Scree Plot



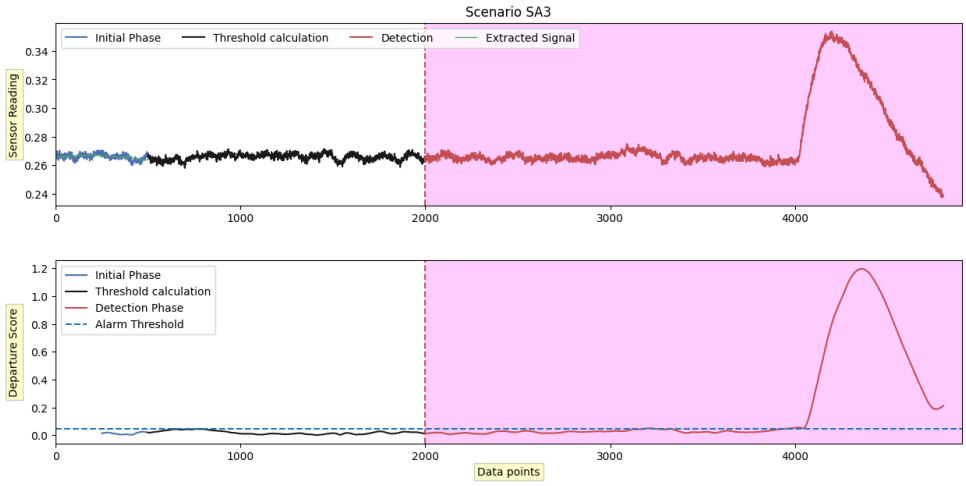
Scenario SA3

SA3- 0^{th} Column

Threshold value $= 0.045315248036012674$

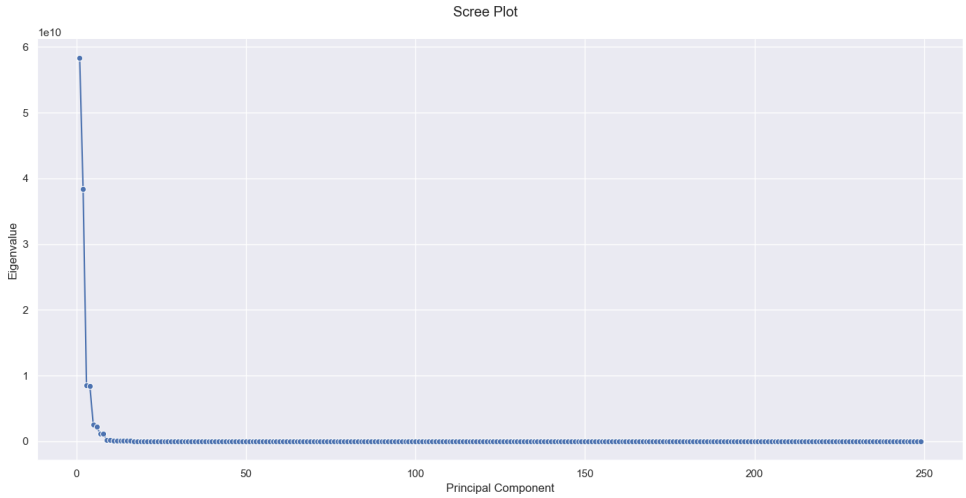
$N = 2000$

First 500 for initial phase and remaining for calculating threshold value.



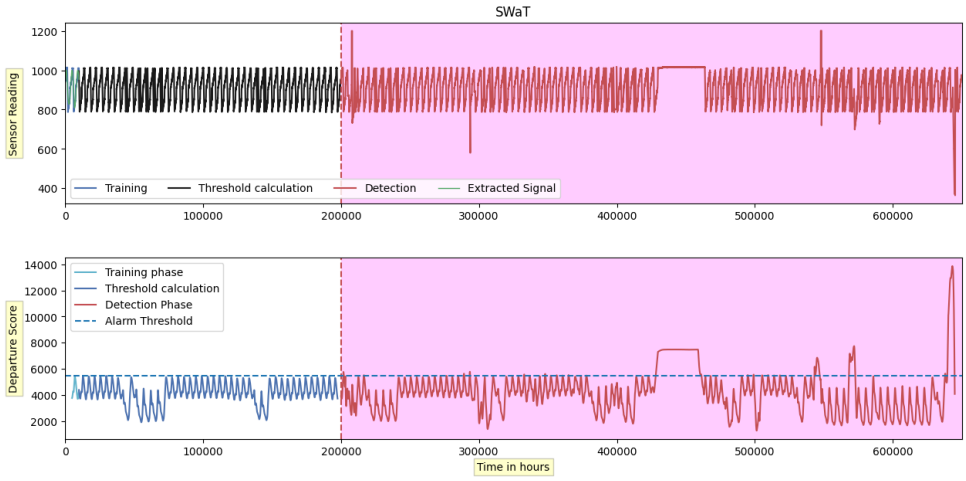
2.5.2 Dataset Analysis and Plotting (SWaT Dataset)

SWaT Scree Plot



Scenario SWaT

SWaT- 14^{th} Column (0-indexed, LIT301)
Threshold value $= 5481.096492451534$
$N = 200000$
First 10000 for initial phase and remaining for calculating threshold value.



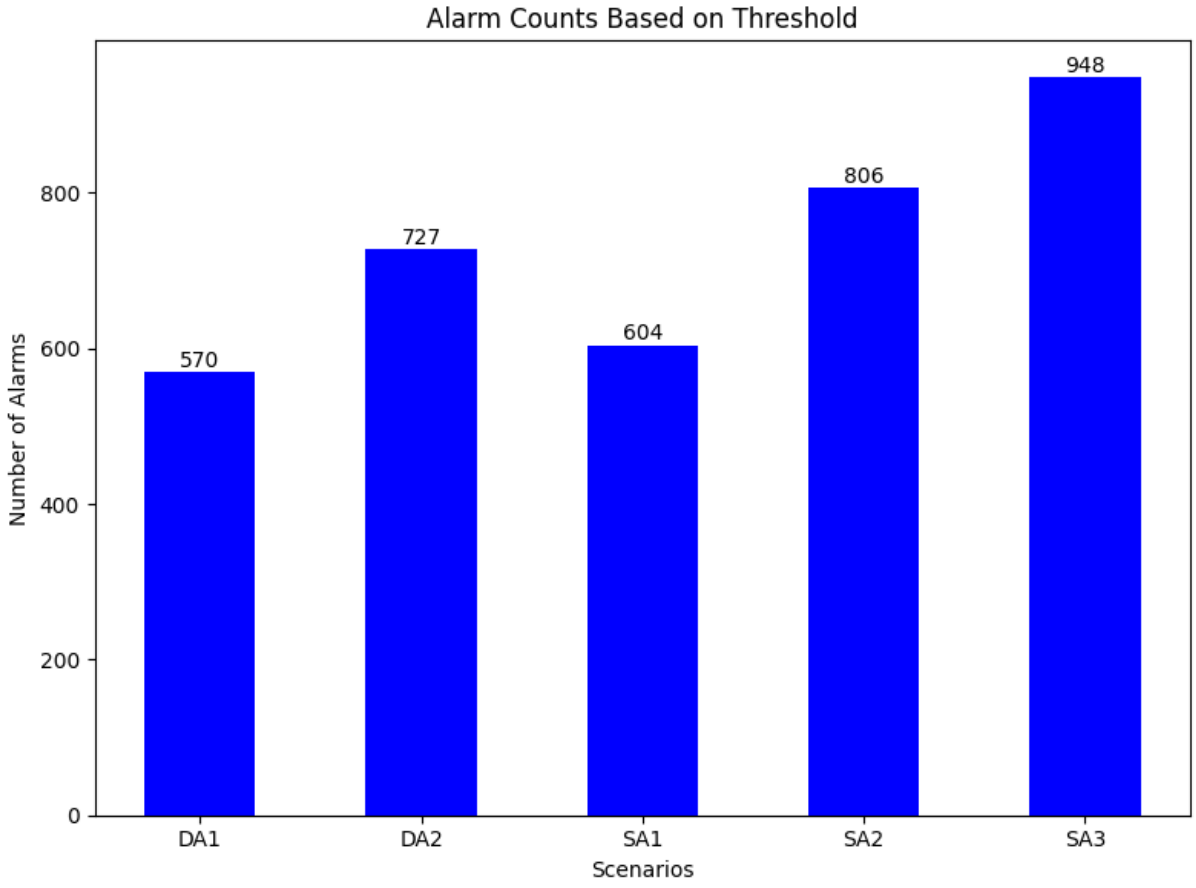
2.5.3 Performance Comparison

Runtime in $P = U\hat{T}$ is slightly higher than $P = UU\hat{T}$ for Scenario DA1 because in the notebook there are some markdown statements for explanation purpose, therefore it takes longer time when it should take less. On removing those markdowns it will take less time than $P = UU\hat{T}$, which took one hour plus time on the SWaT dataset and we had to stop it thereafter. Refer the screenshot below:

Running Time Table

	Scenario DA1	SWaT
P = U^T	0.8756225109100342	358.8855664730072
P = UU^T	0.6664793491363525	inf

2.5.4 Attack Scenario Analysis in TE Dataset



The bar chart displays the alarm counts for five attack scenarios (DA1, DA2, SA1, SA2, SA3) in the TE dataset based on a threshold, as part of PASAD implementation. The x-axis represents the attack scenarios, and the y-axis shows the number of alarms triggered. The highest number of alarms is observed in scenario SA3 with 948 alarms, followed by SA2 (806 alarms) and DA2 (727 alarms). Scenarios SA1 and DA1 have relatively fewer alarms, with 604 and 570 alarms, respectively. These results reflect the sensitivity of PASAD to different attack types while maintaining a zero false alarm rate.

3 Problem 2

The **GitHub repository** containing the detail code can be found **here.**

3.1 Justification

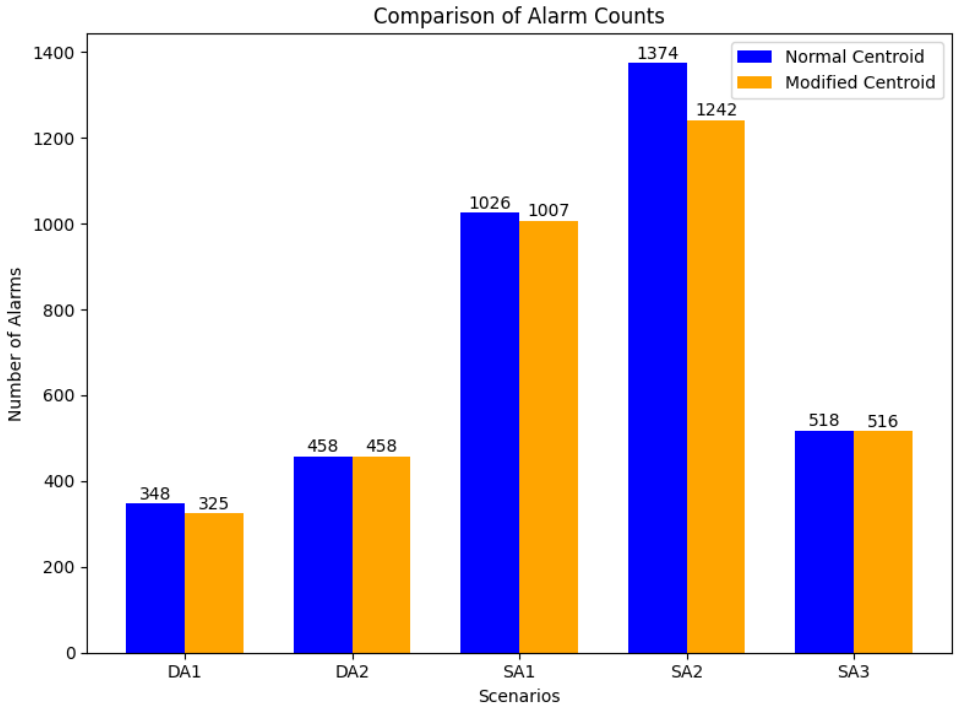
Difference Between Mean and Centroid

Mean: In PASAD, the mean of the cluster typically refers to the arithmetic mean (average) of the lagged vectors used during training.

Centroid: The centroid, in this context, is essentially the geometric center of the set of points in a subspace. It is calculated in the same way as the mean but could have different implications when used for time-series data. Here, we have calculated it as $C = (min(X) + max(X))/2$ (As directed by Prof via e-mail).

The performance degrades when modifying the centroid calculation because the modified approach only takes into account the minimum and maximum values, which are often outliers. This skewed representation ignores the majority of the data points, particularly those that are densely concentrated in a specific region. In many cases, the central tendency of a cluster lies where the data is most densely populated, not at the extremes. By focusing only on the minimum and maximum values, the centroid no longer accurately represents the true center of the data distribution, leading to poorer anomaly detection and an increased likelihood of errors.

3.2 Implementation and Comparison



4 Problem 3

The **GitHub repository** containing the detail code can be found **here.**

4.1 Justification

Euclidean vs Mahalanobis Distance

Euclidean Distance: In PASAD, Euclidean distance calculates the straight-line distance between the projected test point and the cluster center (either the mean or centroid). It assumes that all directions in the feature space are equally important and the data has no correlation across dimensions.

Mahalanobis Distance: Mahalanobis distance, on the other hand, accounts for correlations between features by measuring the distance from the centroid using the covariance matrix. It effectively "scales" the data by taking into account the distribution of the points, giving more weight to directions where the variance is low (i.e., where data points are tightly clustered) and less weight to directions with high variance.

Advantages of Mahalanobis Distance

Sensitivity to Correlation: Mahalanobis distance is sensitive to correlations in the data, which is useful if certain dimensions (lagged vectors in PASAD) are correlated. By accounting for the covariance, Mahalanobis distance can better represent the structure of the data, leading to more accurate anomaly detection.

Better Handling of Non-Spherical Clusters: If the normal data cluster is non-spherical, Euclidean distance may not capture the true shape of the distribution. Mahalanobis distance can handle such cases by scaling according to the data's covariance, potentially improving detection capability.

Noise Reduction: In datasets with noisy dimensions or outliers, Mahalanobis distance can reduce the impact of irrelevant or noisy features by weighting dimensions according to their variance.

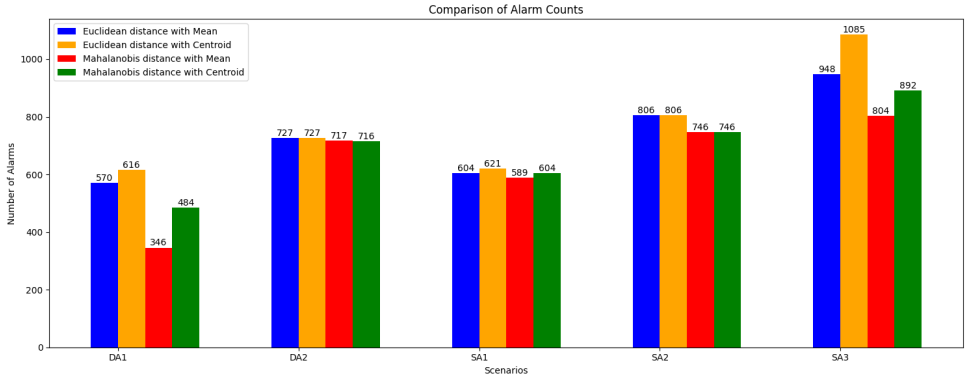
Disadvantages of Mahalanobis Distance

Increased Complexity: Mahalanobis distance requires the computation of the covariance matrix, which adds computational overhead compared to Euclidean distance. This could increase the runtime, especially for high-dimensional data.

Inaccuracy in Low-Volume Data: If the training data is insufficient, the covariance matrix may not be accurately estimated, leading to inaccurate distance calculations and potentially degraded performance. In cases where the data is highly sparse or noisy, the covariance matrix might be ill-conditioned or singular, causing numerical issues.

Overfitting Risk: By tailoring the distance measure to the data's specific covariance structure, Mahalanobis distance may overfit to the normal data and reduce the generalizability of the anomaly detection method

4.2 Implementation and Comparison



This comparison shows that using Mahalanobis distance generally results in fewer alarms, especially when using the mean, suggesting that Mahalanobis distance may reduce sensitivity to anomalies but could also result in better control of false alarms.

4.3 Runtime Analysis

Running Time Table

	Euclidean distance	Mahanabolis distance
Mean	0.0	17514228.82080078
Centroid	0.0	15378475.189208984

It is quite evident from the table that the time taken for euclidean distance is negligible as compared to Mahanabolis distance. Time mentioned in table above is in ns. The Mahanabolis distance takes into account the calculation of Inverse covariance matrix and hence the more time.