# Project 2:

# Time Series

**Report Created by:** Avi Tombak

avitomba@buffalo.edu

**University at Buffalo**

**CSE454 Fall 2021**

11/12/2021

## Table of Contents

# Summary

This project processes a time series dataset, in particular, a deterministic stationary time series of synthetic control data provided by the University of California Irvine. This project is implemented using a Matlab program. The time series data set is processed by using two representation techniques, piece-wise aggregate approximation (PAA) and symbolic aggregate approximation (SAX). Using these representation techniques, classification on the dataset samples is performed. These classifications are done using Euclidean and Manhattan distance methods. These classification results are then analyzed via confusion matrices.

*Keywords: Time Series, Dataset, Sample, Piece-wise Aggregate Approximation, Symbolic Aggregate Approximation, Representation, Classification, Euclidean Distance, Manhattan Distance, Confusion Matrix, Matlab*

# Dataset

The dataset used in this project is provided by the University of California Irvine.

https://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series

It contains six-hundred samples of time series representations for synthetically generated control charts. These samples are classified into six distinct classifications, normal, cyclic, increasing trend, decreasing trend, upward shift, and downward shift. This dataset is stored in an ASCII file with six-hundred rows and sixty columns, where each row signifies a sample in the dataset, and each column signifies an observation within a sample.

# Process for Working with Time Series

The process for working with time series datasets draws many parallels to the engineering design process. In particular, this process can be broken down into five distinct steps. Figuring out the problem, reading in the data, data preprocessing, data sampling and representation, and data analysis. Each step can be summarized as follows:

1) Figure out the problem

- Derived from project handout
  - Explore time series representation and classification behaviors
  - Utilize the synthetic control dataset

2) Read in the data

- This can be done utilizing a numerical matrix in a Matlab program, in particular one of size 600x60 for this dataset

3) Data preprocessing

- The data points have values within a range of zero to one-hundred
- Normalization can be used to scale these values within a range of zero to one

4) Data sampling and representation

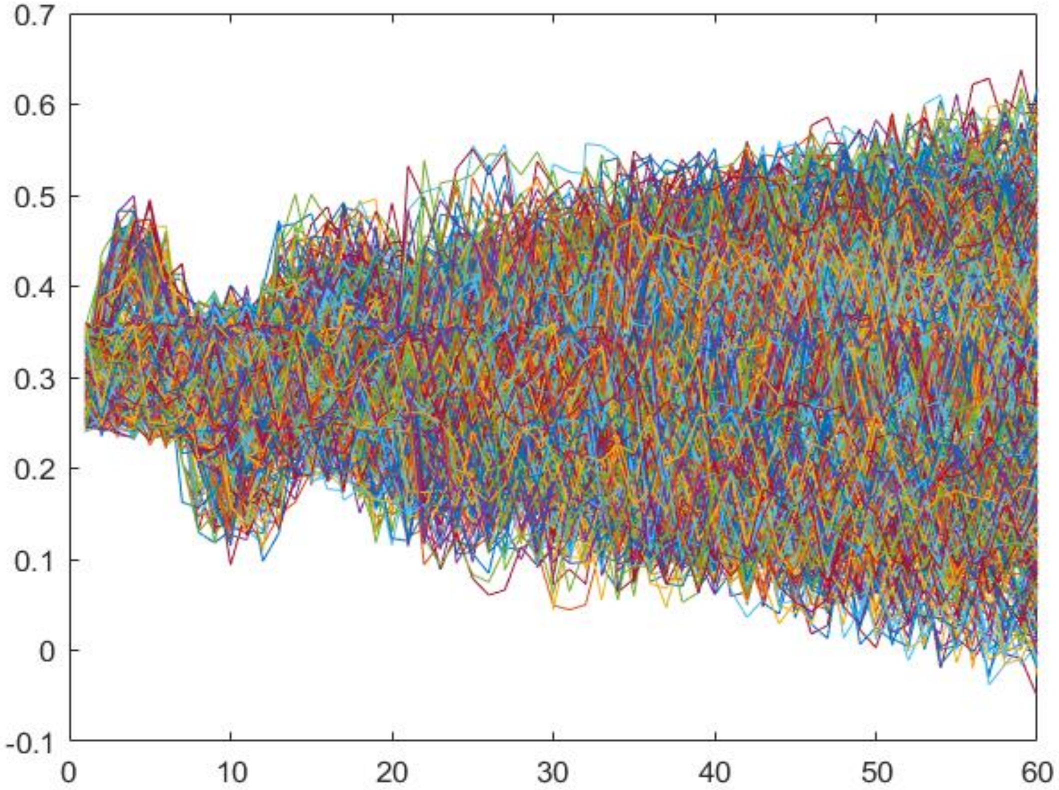
- Two representation techniques are to be used, PAA and SAX
- Data can be sampled in order to establish a training and testing set

5) Data analysis

- Classification and be done on the created representations using distance methods, in particular both Euclidean and Manhattan.
- Classification accuracy can be measured with a structure known as a confusion matrix.

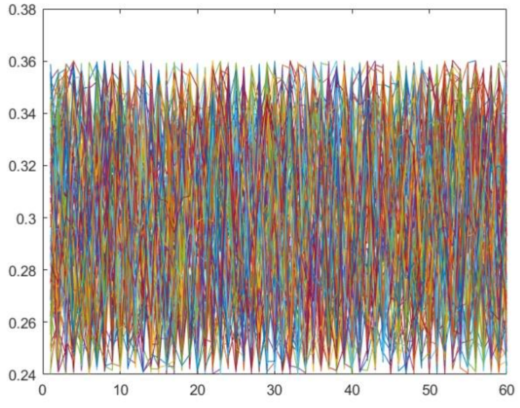
# Original Time Series Plots

**Plot of all six-hundred samples in the dataset:**

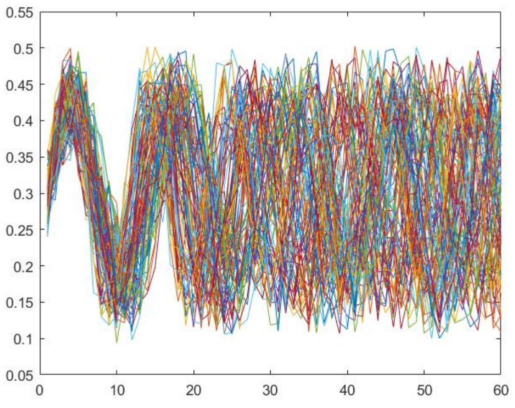


**Plot of each distinct class and its one-hundred samples:**

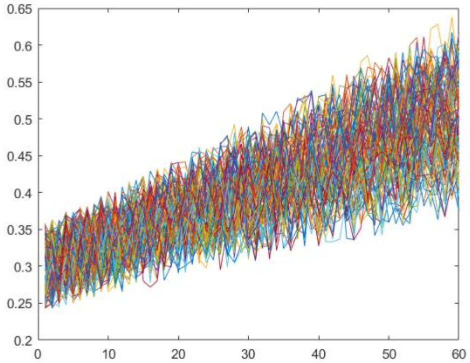
Normal:



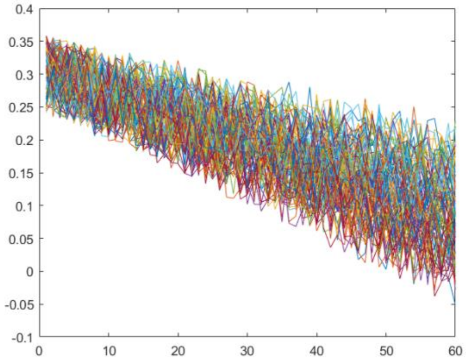
Cyclic:



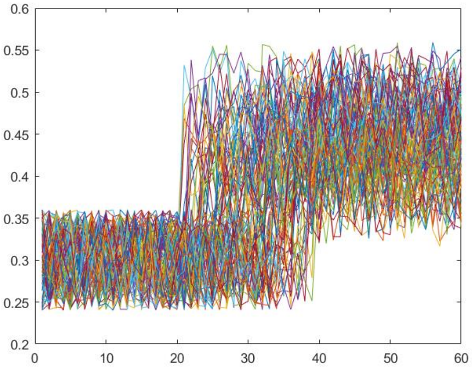
Increasing Trend:



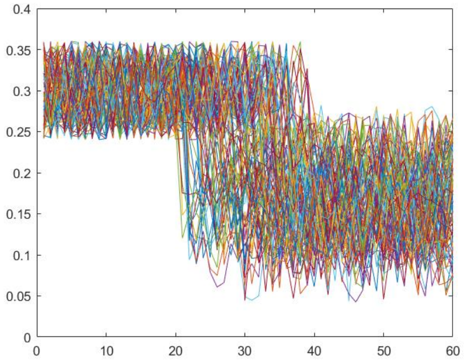
Decreasing Trend:



Upward Shift:



Downward Shift:



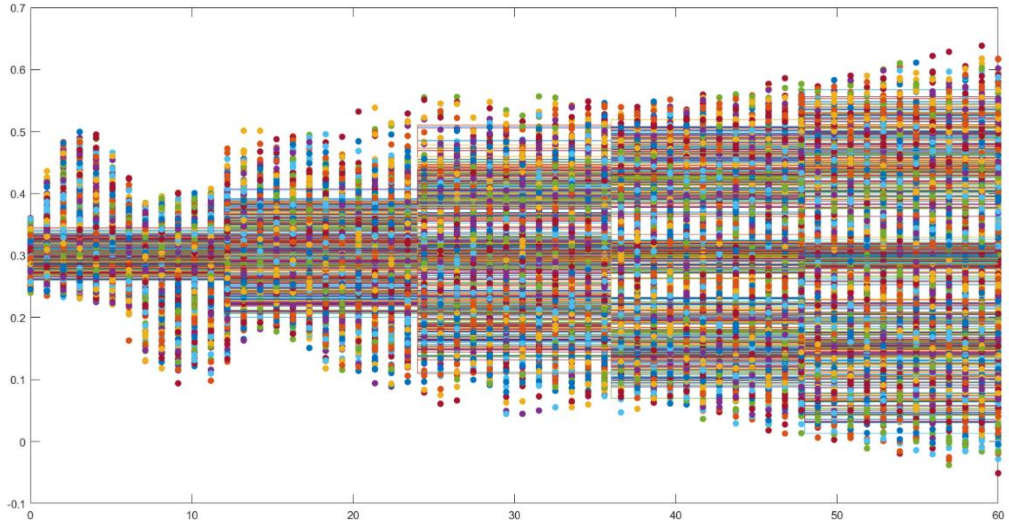
## PAA Approach and Implementation

Piece-wise aggregate approximation (PAA) is a representation technique that reduces data based on aggregate values within piece-wise regions corresponding to a time series. It requires the use of expert knowledge of the dataset, including the number of samples and number of observations for the dataset. With this expert knowledge and segmentation of the time series, the samples can be simplified with an approximation of each segment.

In this implementation, the segment count chosen is five, meaning that for the sixty observations within a sample, every twelve is aggregated into a distinct piece-wise region. PAA is applied to the overall dataset, each class's corresponding data subset, and to a mean sample of each class.

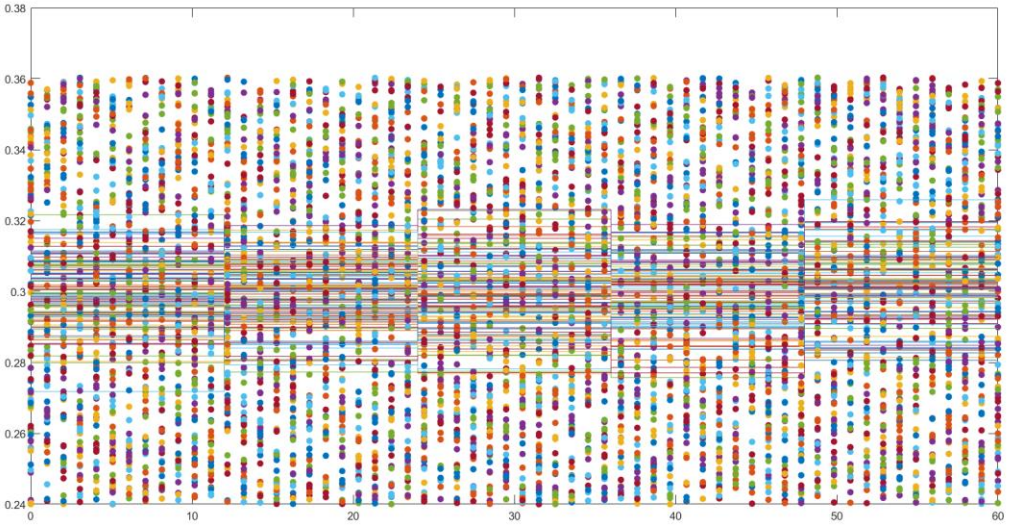
# PAA Plots

**PAAs of Original Dataset:**

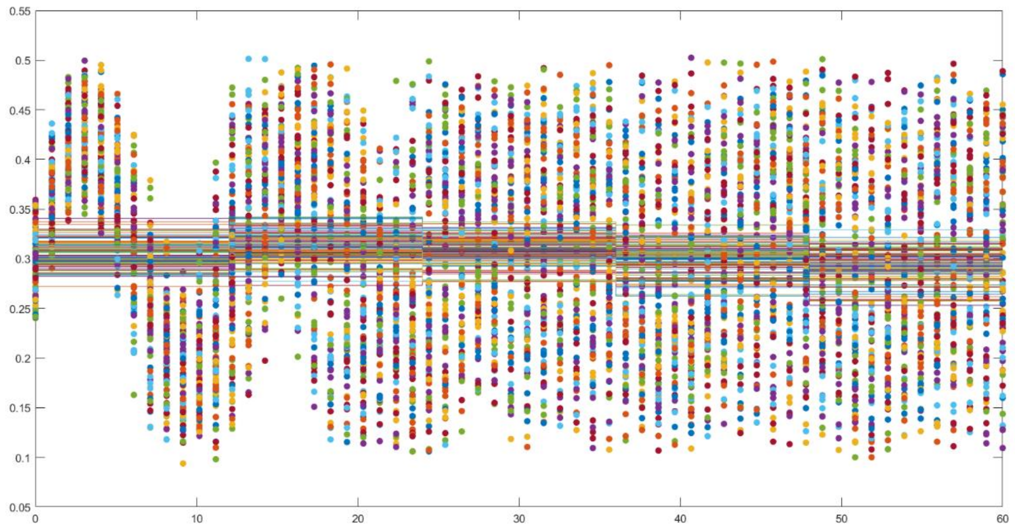


**PAAs of Class Subsets:**

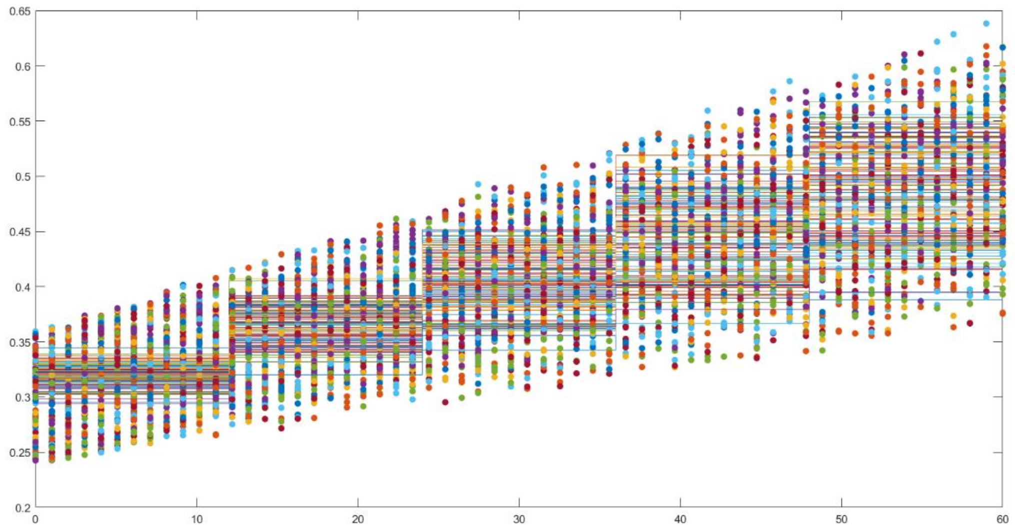
PAAs of Normal:



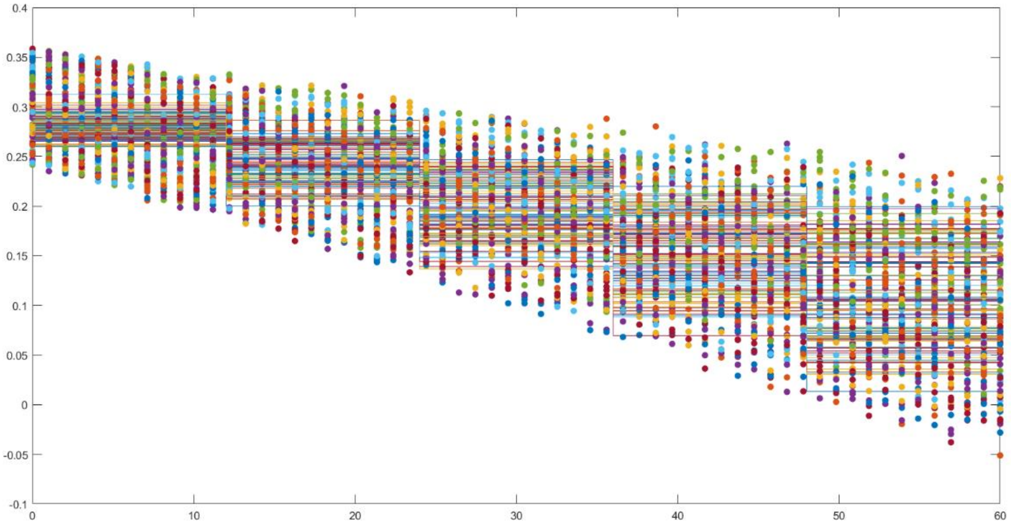
PAAs of Cyclic:



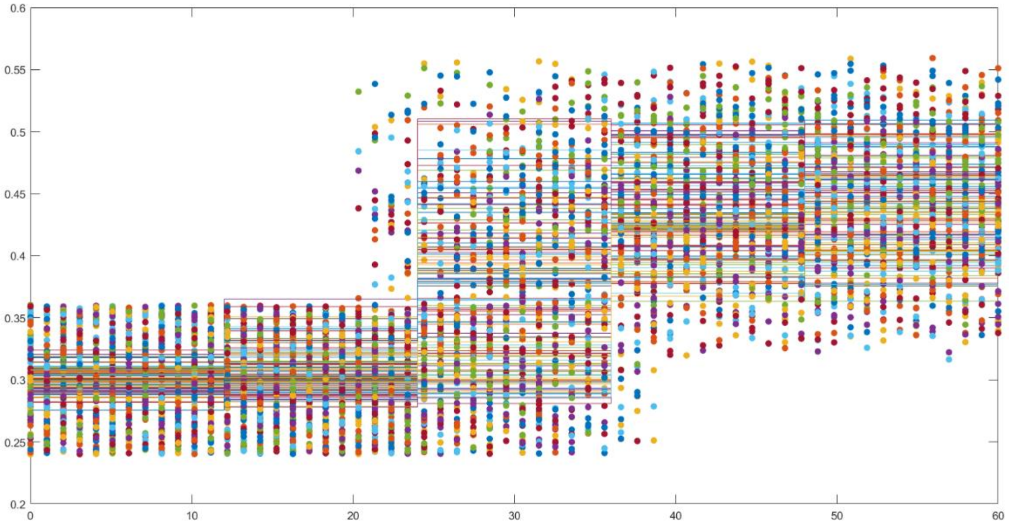
PAAs of Increasing Trend:



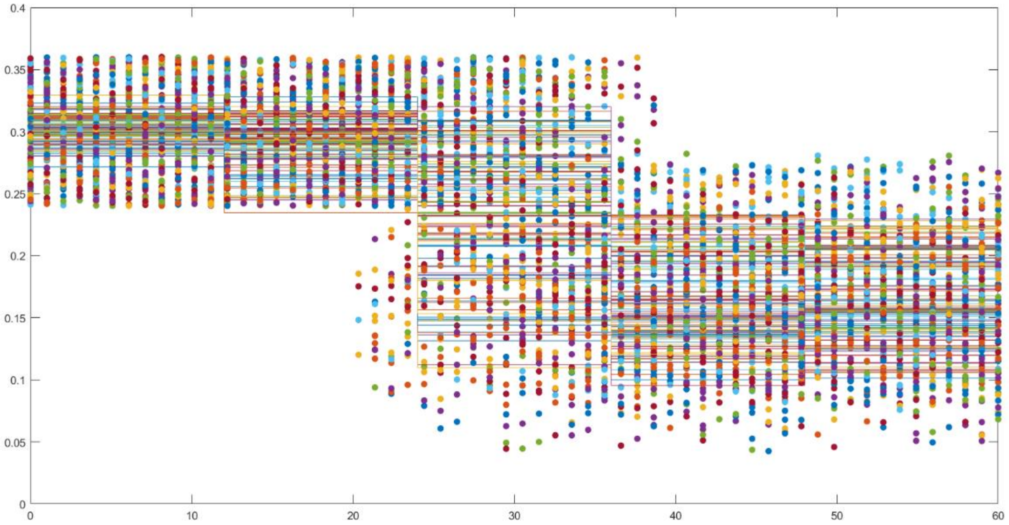
PAAs of Decreasing Trend:



PAAs of Upward Shift:

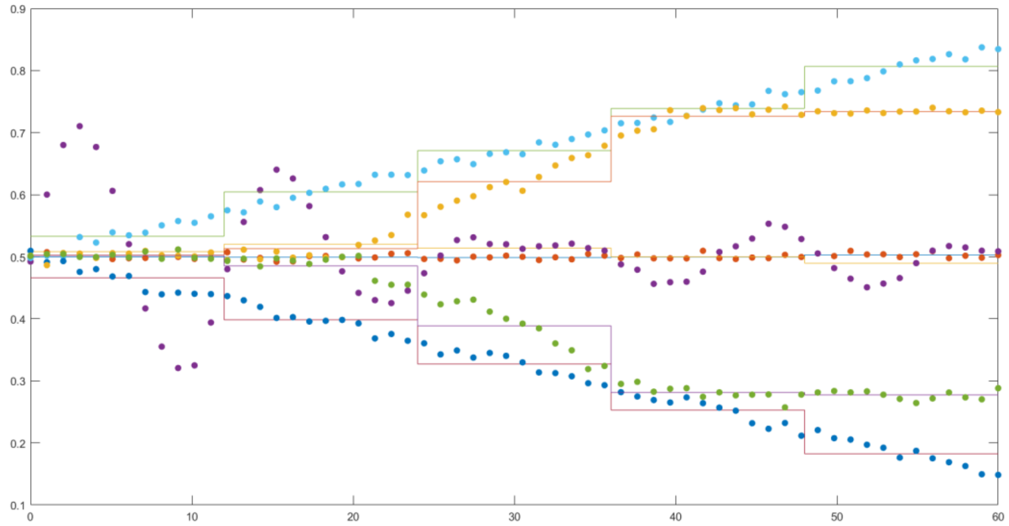


PAAs of Downward Shift:

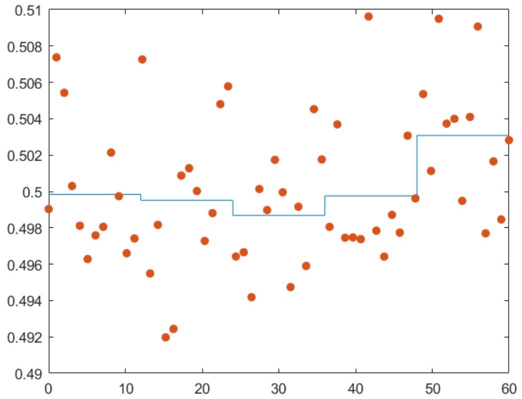


**PAAs of Mean Class Samples (Testing Data):**

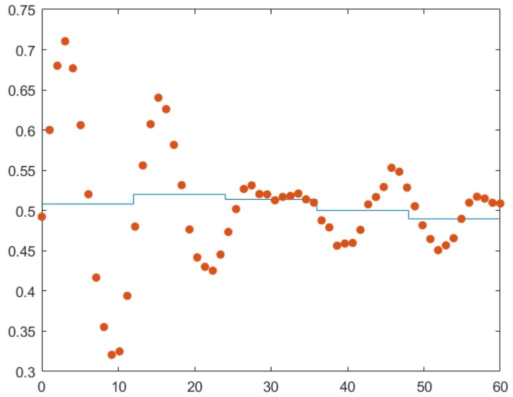
PAAs of Testing Set



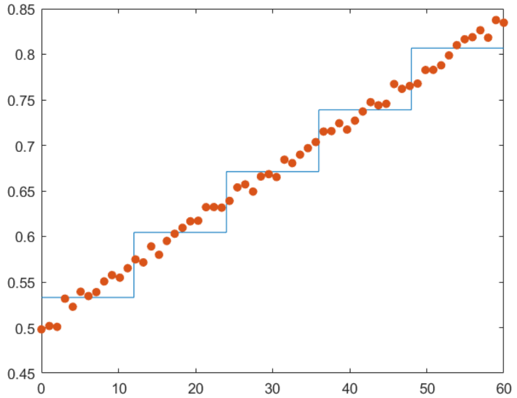
PAA of Normal Mean:



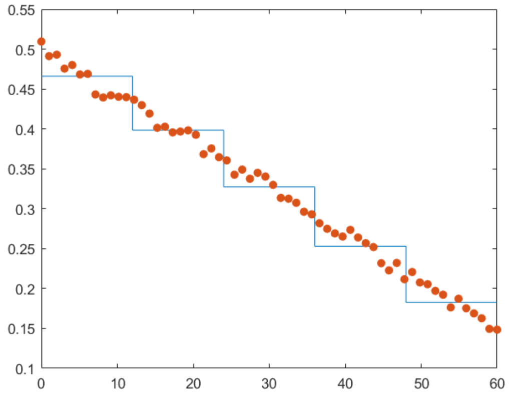
PAA of Cyclic Mean:



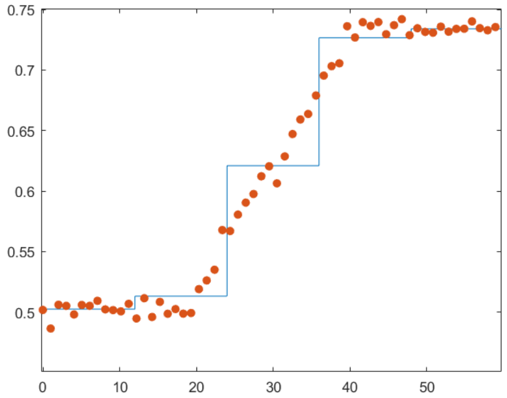
PAA of Increasing Trend Mean:



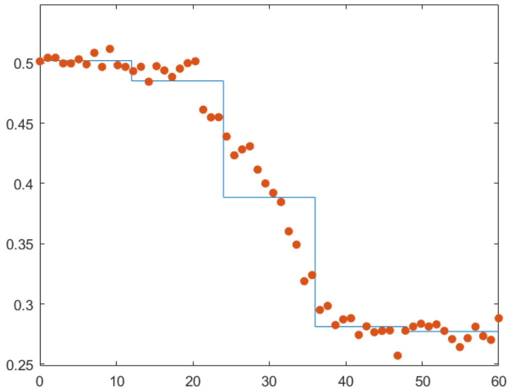
PAA of Decreasing Trend Mean:



PAA of Upward Shift Mean:



PAA of Downward Shift Mean:



## SAX Approach and Implementation

Not Available.

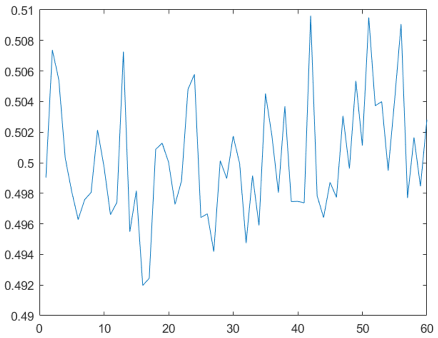
## SAX Plots

Not Available.

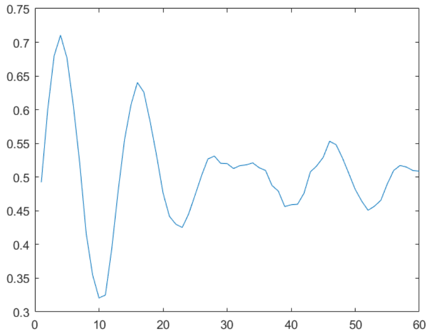
## Training and Testing Data Generation

Since classification processing was to be done on the entire dataset, the original dataset was used as the training data. The testing dataset has six samples, one example of each class, that example being the mean sample of the class's corresponding subset. The plots of the testing dataset are as follows:

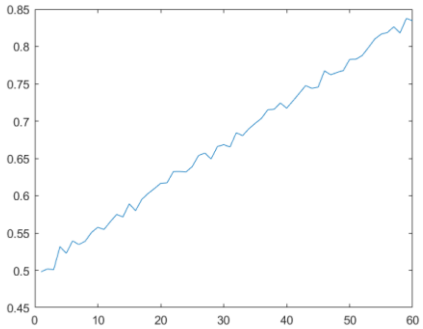
Normal:



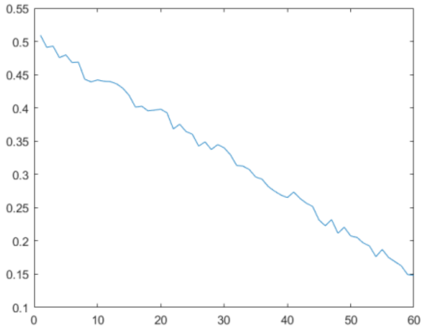
Cyclic:



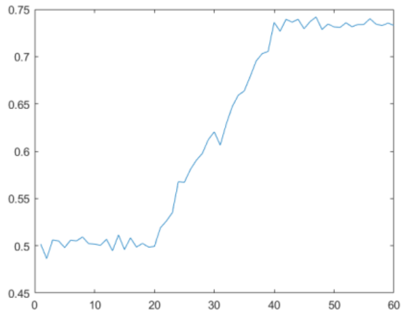
Increasing Trend:



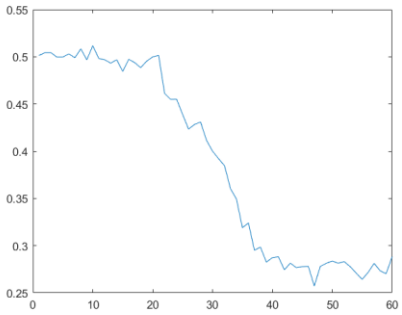
Decreasing Trend:



Upward Shift:



Downward Shift:



## Classification Process

The original dataset had six distinct classes, each with one-hundred samples. With this time series processing, classification inferences can be made based on comparing the training data and testing data. This can be done by comparing the distance between training samples and testing samples. Distance can be calculated in a variety of methods, in this project Euclidean distance and Manhattan distance were used.

Euclidean: $d= \surd(\Sigma((s[i]-r[i])^{2}))$

Euclidean: $d= \Sigma( |(s[i]-r[i])| )$

where s and r are two time series or samples,

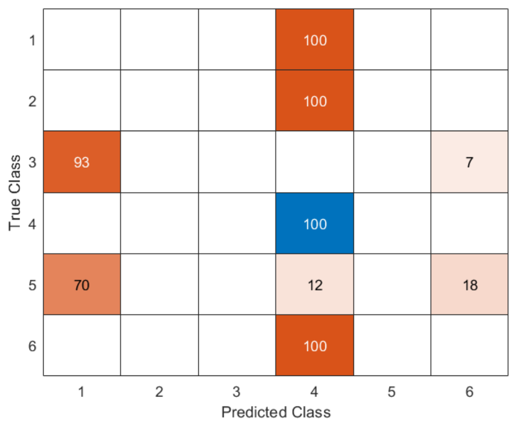
In this case one from the training set, one from the testing set.

Every sample from the training set is classified with the same label as the closest distant sample from testing set. With this method, the time series processing can predict the classes of the training set samples.

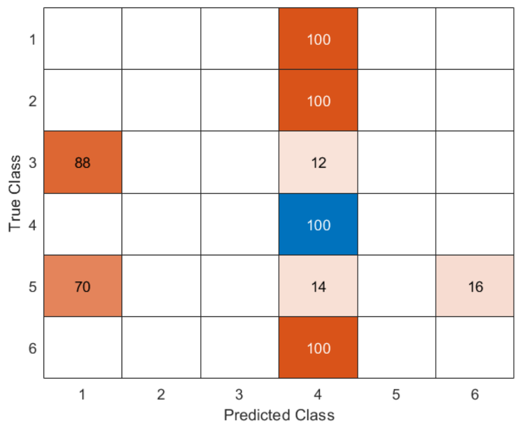
## Results

Classification on the original dataset using the testing dataset and distance methods was performed. This process had egregiously poor accuracy, of only 16.67% correctness for both Euclidean and Manhattan distances. This points to an obvious error or oversight within the implementation, and this error was not fixed. The following confusion matrices were generated corresponding to each distance method. Notice the concentration of class 4 (decreasing trend) predictions, this is the error in question.

Euclidean Distance:



Manhattan Distance:


True Class
Predicted Class