# The Hong Kong University of Science and Technology

## MSBD5002 Data Mining and Knowledge Discovery

## Group 29 Report

## Multi-dataset Time Series Anomaly Detection

*HE Shuo 20797403 sheaw@connect.ust.hk*
*ZHU Zhenyi 20784183 zzhubh@connect.ust.hk*
*FENG Geqin 20787575 gfengac@connect.ust.hk*

# 1. Introduction

The task of this project is to find the center of an anomaly in each test-data set of total 250 time-series. Firstly, we do some preparing works like data loading and data preprocessing. Secondly, we implement our models to get the transformed data we want for subsequent processing. In this part, we divide our models into 3 types, including basic statistical models, general time-series detection models and deep learning models. Thirdly, we integrate the data from different models and try different ensemble methods, and finally choose one to generate the final results.

# 2. Preparing works

## 2.1 Data loading

In the code of each model, we use *pandas* related APIs to load the data. Some data sets need to be paid attention because the interval between data in them is different from others. Then, after loading all the data sets, we transform them into *np.array()* to make subsequent operations more rapidly and cpu-friendly by using *numpy* APIs.
The exact function name of Data loading in each model is “dataLoad”.

## 2.2 Data preprocessing

### 2.2.1 Feature Extraction

We use *data_info.py* to get the length and partition (the separation point of training data and test data) and save them by *np.savetxt()*.

### 2.2.2 Period Extraction

Since some models (Matrix Profile, Averaged Sliding Window, etc.) need the period information of data sets, an algorithm is referenced and improved by us. And we will describe it below:
(1) Iterate over all integers from *1* to *0.04 * total length of the data set* as the gap distance *d*.

(2) Find all the peak values (maximum values, *pd*) the valley values (minimum values, *vd*) in each gap distance.
(3) Calculate the period score as:

$$S_d = \frac{min(std(pd), std(vd))}{\sqrt{d}}$$

(4) Repeat (2) and (3) until find the smallest period score, and finally we set *d* with the smallest $S_d$ as the period as the data set. Thus, we could find periods for all 250 data sets.

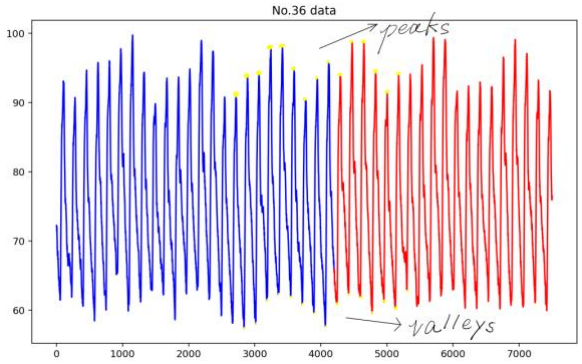



Figure 1: The picture shows some peaks and valleys for data set 36.

### 2.2.3 Anomaly Classification

We divide the anomaly into 2 types: point anomaly and collective anomaly. And different approaches could find different types of anomalies, we will introduce them later.

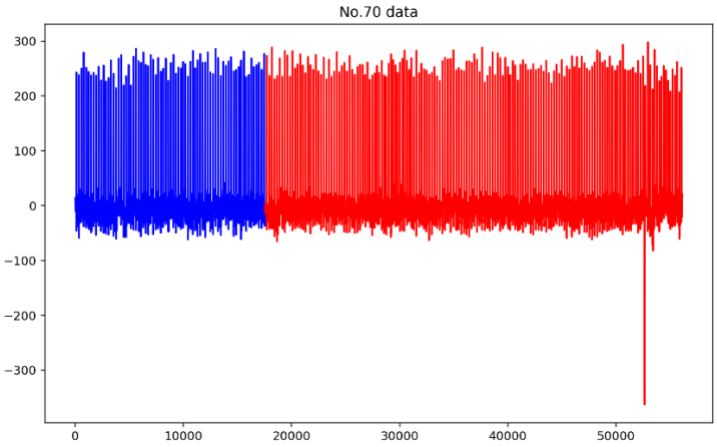



Figure 2: point anomaly in data set 70

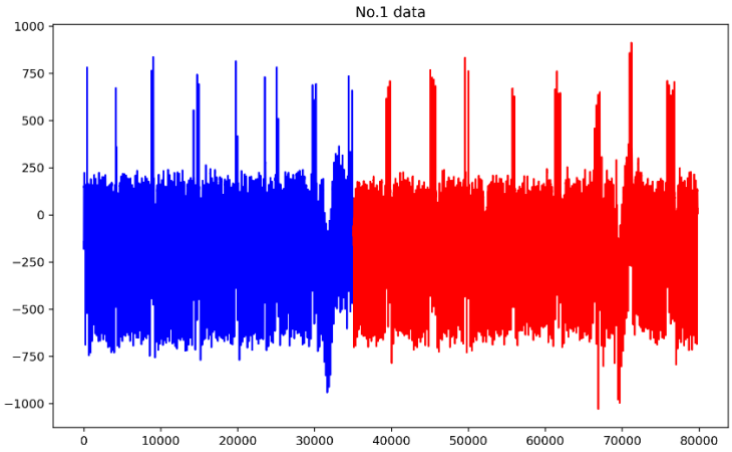



Figure 3: collective anomaly in data set 1

# 3. Approaches

## 3.1 Basic statistic models

We firstly try to use statistic approaches based on original curves, 1-order curves, and 2-order curves to find the anomaly[1]. Next we will introduce some details about our approaches.

### 3.1.1 The local range based on original, 1-order, and 2 order curves.

We calculate 3 types of ranges on a selected interval as showed below. The interval ranges of original curves reflect the intuitive changes of the data. The interval ranges on 1-order curves reflect the changes of growth rates of the data. And the interval ranges on 2-order curves reflect the changes of the changes of the growth rates of the data, which mean a deeper derivation of the data. Three types of ranges are shown as below:

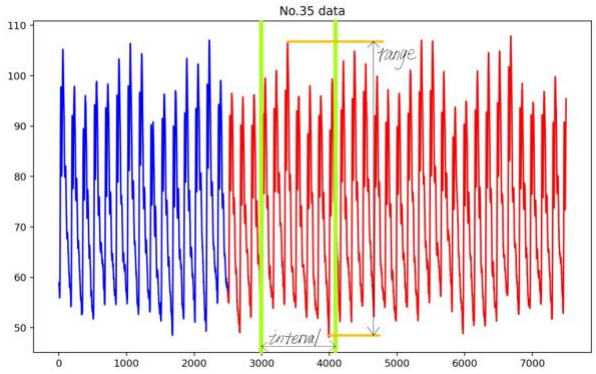



Figure 4: original local range in data set 35

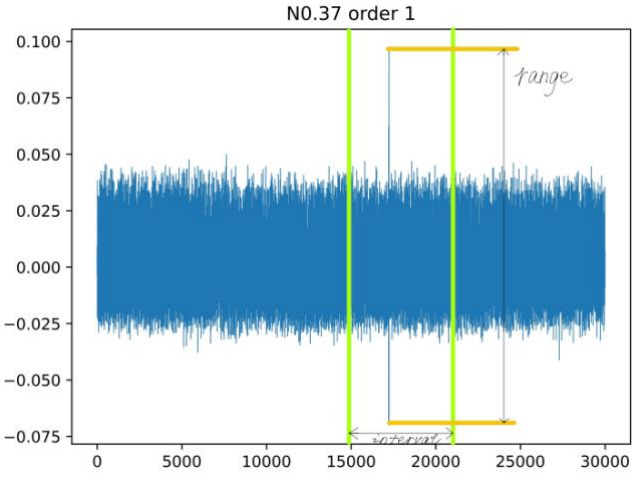



Figure 5: 1-order local range in data set 37

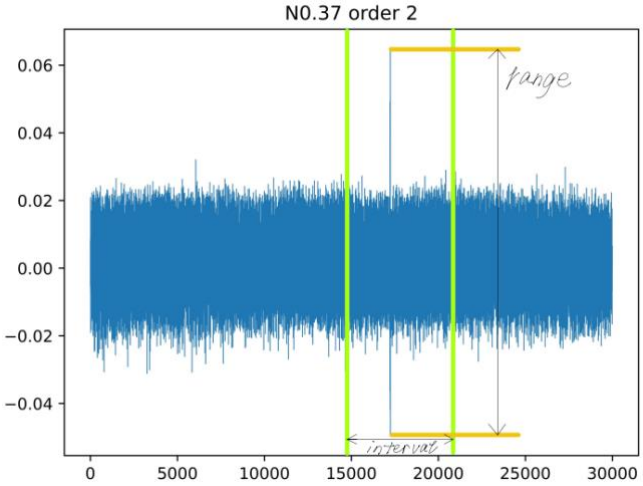



Figure 6: 2-order local range in data set 37

### 3.1.2 The local mean based on 1-order curves

We calculate the local mean on 1-order curves to get the smoothing changes of growth rate. We calculate on 3 types: Filtering low absolute values to get the high extremum disturbances, filtering high absolute values to get the low extremum disturbances, and dotting 1-order curves with themselves to magnify the overall trend. The schematic diagrams are as follows:

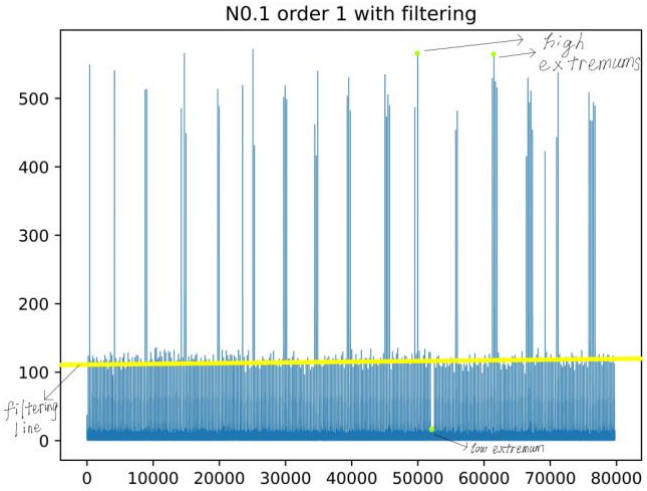


Figure 7: 1-order local mean based on filtering in data set 1

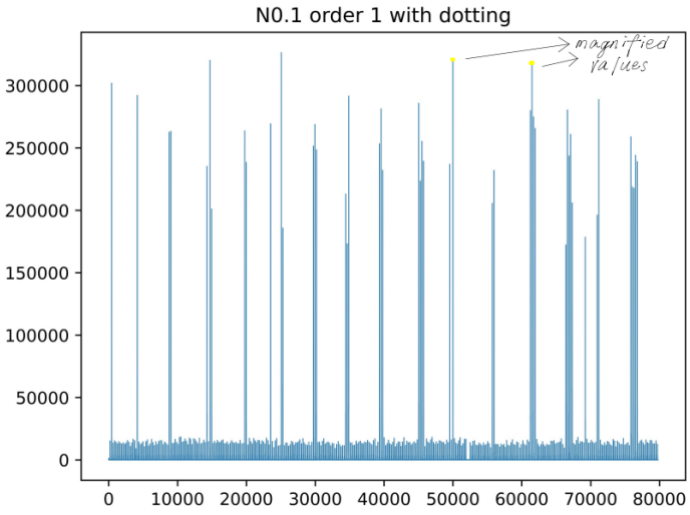


Figure 8: 1-order local mean based on dotting in data set 1

### 3.1.3 The local standard deviation based on 2-order curves

We calculate the local standard deviation based on 2-order curves to observe the degree of discrete of 2-order calculation results, this will help us to find the abnormal disturbances of the curves. The schematic diagram is as follow:

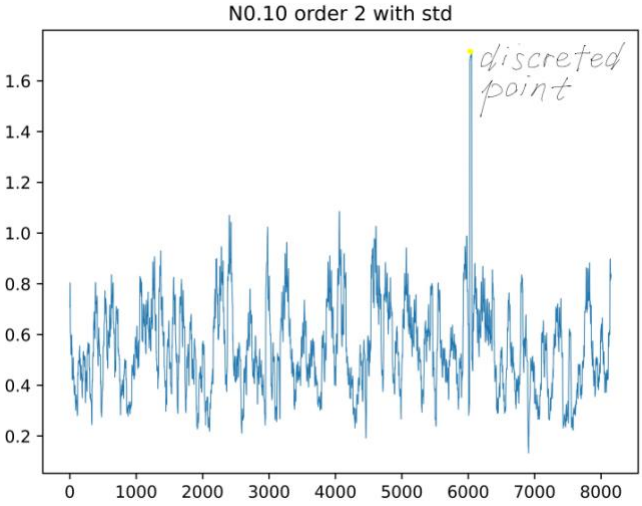


Figure 9: 2-order local std based on dotting in data set 10

### 3.1.4 The inverse curves

We also calculate the inverse curves of the local range based on original curves and the local standard deviation based on 2-order curves by filtering the data with a large rate of changes, so we can leave the disturbances with a low rate of changes. For some data sets, this approach is efficient to help us to find the anomaly. The schematic diagrams are as follows:

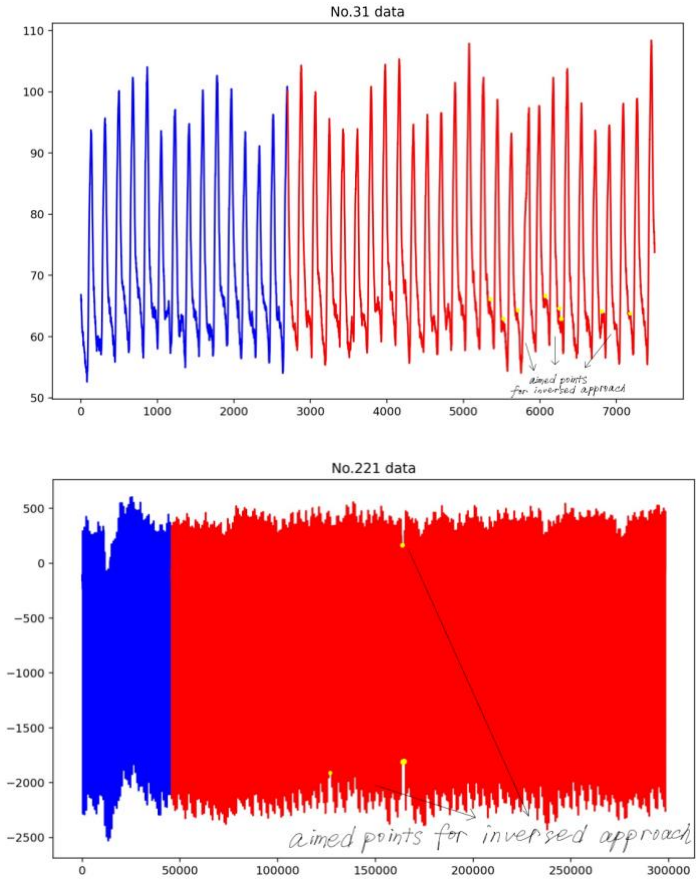



Figure 10: aimed points of inversed method in data set 31 and 221

### 3.1.5 Selection method

For each statistical model, we set a range of period to select the best result, the criterion will be showed in the ensemble part. We set the range from *20* to 0.05 * *0.04* * *total length of the data set*.

### 3.1.6 Pros & Cons of statistical approaches

Most of the statistical approaches are sensitive to point anomalies, but are not so sensitive to collective anomalies.

## 3.2 Model based on Fast Fourier Transformation

### 3.2.1 Introduction

In this approach, we transform the time series in time domain to its corresponding frequency domain. Then we apply a low pass filter which eliminates all frequencies above the cutoff frequency while passing those below the cutoff unchanged. After filtering, the time series will also be transformed back to time domain[2]. Four figures are presented for each time series in the dataset, including both the time and frequency domain signal plots before and after filtering.

### 3.2.2 Result

For example, the figures for "095_UCR_Anomaly_4000.txt" are shown below.

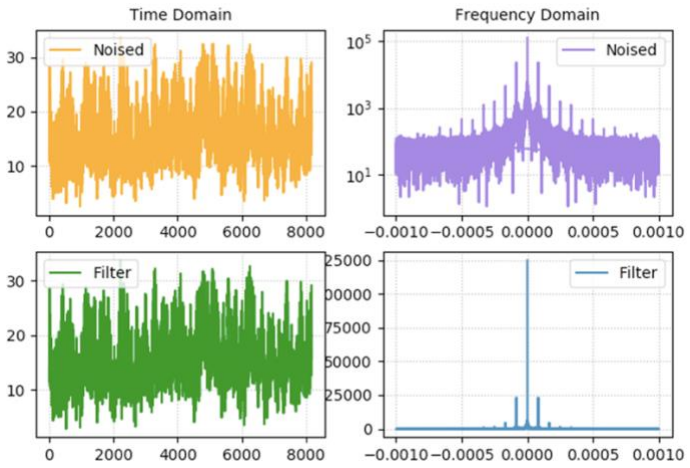



Figure 11: FFT on data set 95

The transformation can eliminate the anomalies in the signal. Different combinations of sampling rates and cutoff thresholds are applied to find the best low pass filter for each time series. The difference between the original signal and filtered signal is recorded as the residual, which reveals the positions of anomalies. As is demonstrated below, the residual figure of "095_UCR_Anomaly_4000.txt" shows an obvious position of anomaly.

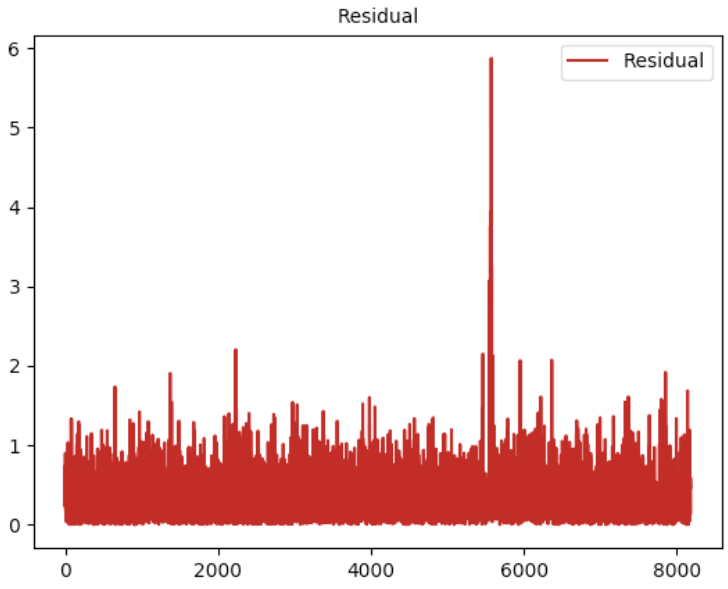



Figure 12: FFT residual on data set 95

### 3.3.3 Pros & Cons of FFT Algorithm

FFT method is able to capture non-repetitive patterns from the time series, which has an advantage over other methods to reveal the anomalies from another respect. On the other hand, since FFT focus on the frequency and periods, it sometimes ignores some fundamentally obvious anomalies. However, this disadvantage can be made up by other methods in our project.

## 3.3 Model based on Spectral Residual

### 3.3.1 Introduction

We observe that many data sets have vibration signals, and we search and find out that SR approach is an efficient way to get the vibration anomaly.

The Spectral Residual (SR) algorithm consists of three major steps: (1) Fourier Transform to get the log amplitude spectrum; (2) calculation of spectral residual; and (3) Inverse Fourier Transform that transforms the sequence back to spatial domain. The specific formulas are shown as follow (given a sequence of **x**):

$$A(f) = Amplitude(\Im(\mathbf{x})) \quad (1)$$

$$P(f) = Phrase(\Im(\mathbf{x})) \quad (2)$$

$$L(f) = log(A(f)) \quad (3)$$

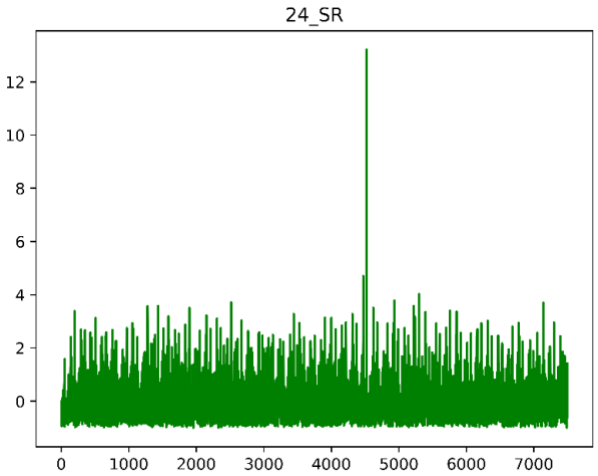
$$AL(f) = h_q(f) \cdot L(f) \quad (4)$$

$$R(f) = L(f) - AL(f) \quad (5)$$

$$S(\mathbf{x}) = \left\| \Im^{-1}(exp(R(f) + iP(f))) \right\| \quad (6)$$

In practice, we use *sranodec* package to generate the anomaly scores. We set the *amplified window size* as *0.5 * selected period*, *series window size* as *0.5 * selected period*, and *score window size* as *2 * selected period*. Then with this three window size, we get the spectral output by the mode of *Silency*, and finally get the anomaly scores.

### 3.3.2 Result



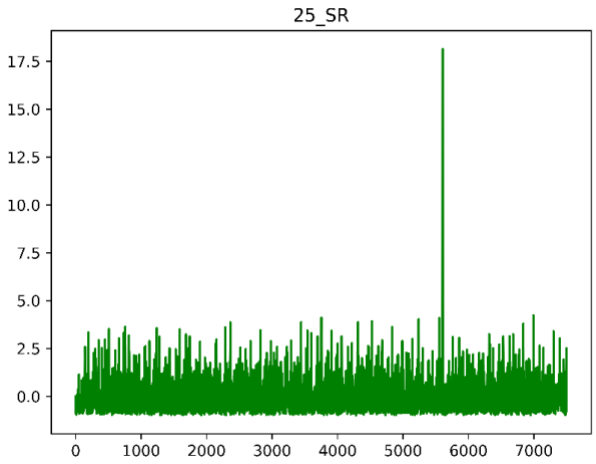



Figure 13: SR on data set 24 and 25

We can see in the above figures, the vibration anomaly could be obviously found.

### 3.3.3 Pros & Cons of SR algorithm

SR algorithm is sensitive to vibration anomalies, but it is limited by the robustness of the selection of the window size, so maybe improper window size will lead to improper results.

## 3.4 Model based on Matrix Profile

### 3.4.1 Introduction

The Matrix Profile is based on the concept of the Distance Profile, if we take a window with a selected size from the original time series data, and slide it along the rest of data, we calculate the Euclidean distance between the window with data at current position, building up the window's Distance Profile. The schematic diagram is as follow:


Figure 12: Process of MP

There are 2 important concepts in MP is motif and discord. In general, motif is a repeated pattern in time series and discord is an anomaly. In the result, motifs will get low values and discords will get high values. In practice, we get SCRIMP++ algorithm to find the discords.

### 3.4.2 Result

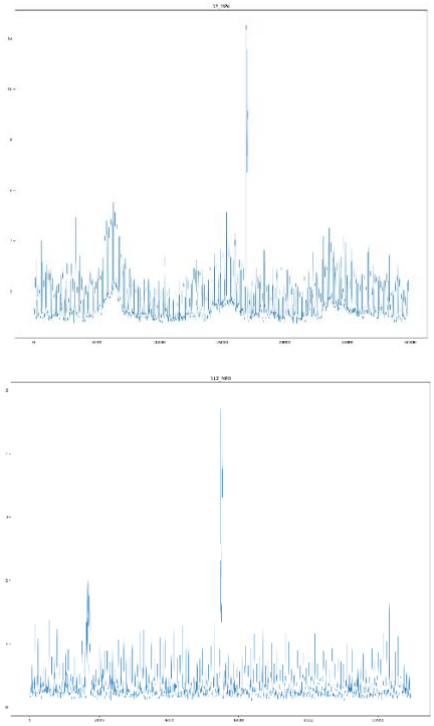


Figure 13: MP on data set 17 and 112

We can see in the above figures, the discords could be obviously found.

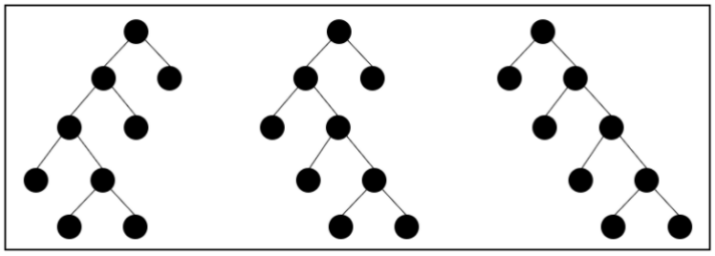
### 3.4.3 Pros & Cons of MP algorithm

For time series, MP algorithm is easily to find the point and collective discords (anomaly), but it also limited by the robustness of the selection of the window size.

## 3.5 Model based on Robust Random Cut Forest

### 3.5.1 Introduction

RRCF is an unsupervised anomaly detection model based on Isolation Forest. It used tree structure displacement to find anomaly and has shown great effect on suddenly changed situation. It uses an ensemble of random forests and trees graphs. In practice, we use *rrcf* package to build forests for each dataset. We set the *nums of trees* as 100, *shingle size* as *0.5 * period*, and *tree size (point nums)* as 4096. *Shingles* result in the anomaly detection running over vectors of data rather than single data points, which smooths the data for minor fluctuations. We collect the *average codisps* to get the anomaly, where, the *codisp* means that when adding a new point, the degree of a RRCT will change[3].



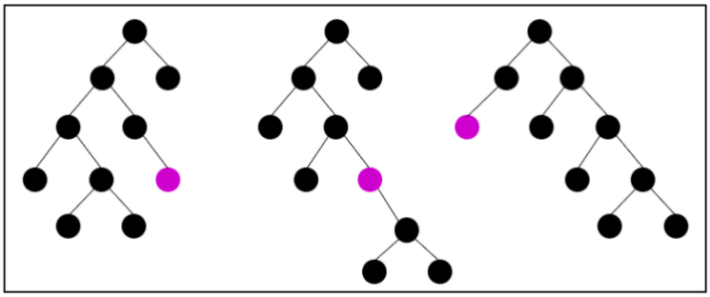


Figure 14: Tree 1 and 3 are significantly changed, but 2 is not.

### 3.5.2 Result

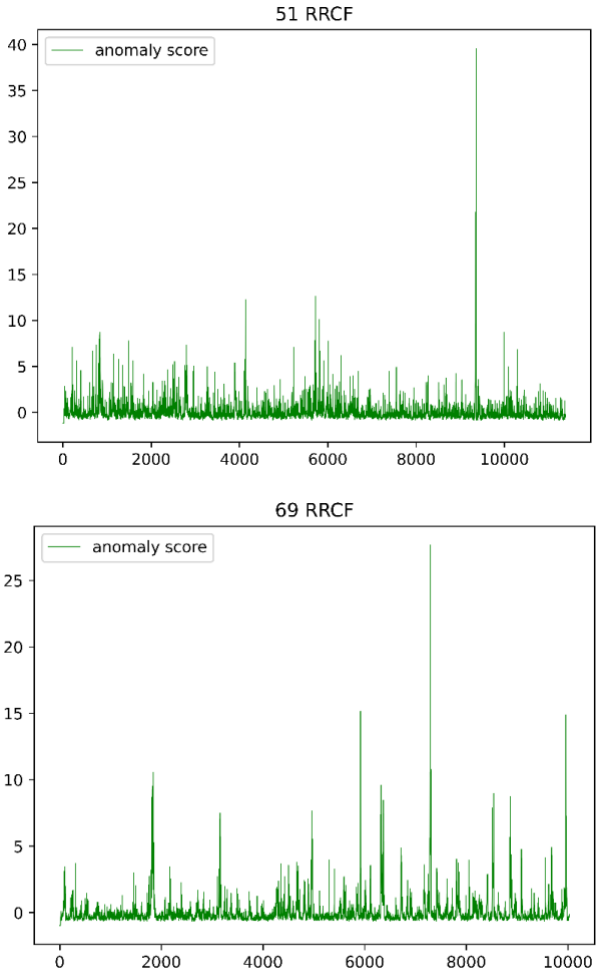



Figure 15: RRCF on data set 51 and 69.

We can see in the above figures, the anomalies could be obviously found.

### 3.5.3 Pros & Cons of RRCF algorithm

RRCF is time-consuming and its efficiency is limited by the tree size. But its result could find the anomalies more robustly.

## 3.6 Model based on Averaged Sliding Window

### 3.6.1 Introduction

Averaged Sliding Window is a common approach to find the anomaly, it is easy to implement and sometimes efficient. It gets the average in a window size and then composes a new feature. The feature are mostly stationary and could enlarge the anomaly.

### 3.6.2 Result

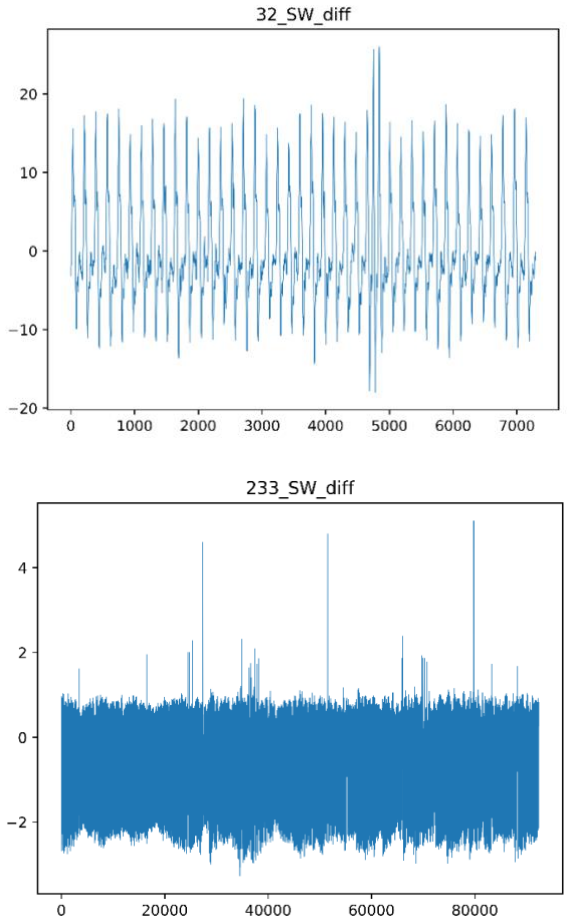



Figure 16: SW on data set 32 and 233.

We could see that in some circumstances, SW is efficient.

### 3.6.3 Pros & Cons of SW algorithm

It is really fast and intuitive, but it only works for obvious tasks.

## 3.7 Model based on Variational Mode Decomposition

### 3.7.1 Introduction

VMD algorithm is an adaptive, completely non-recursive method of modal variational and signal processing. Through it, we could get the main mode of the time series and find anomalies on it. Compared with EMD, the decomposed mode numbers could be decided by user, and it contains more actual meaning[4].

### 3.7.2 Mode number finding (Preprocessing)

We set a range of 1 to 10, and let decomposition number $k$ separately equal to them. With each $k$, we could get $k$ central frequencies (omegas), and we calculate the *kurtosis* of each group of omegas. Finally, we use $k$ with the smallest non-negative *kurtosis* as the decomposition number[5].

### 3.7.3 Result

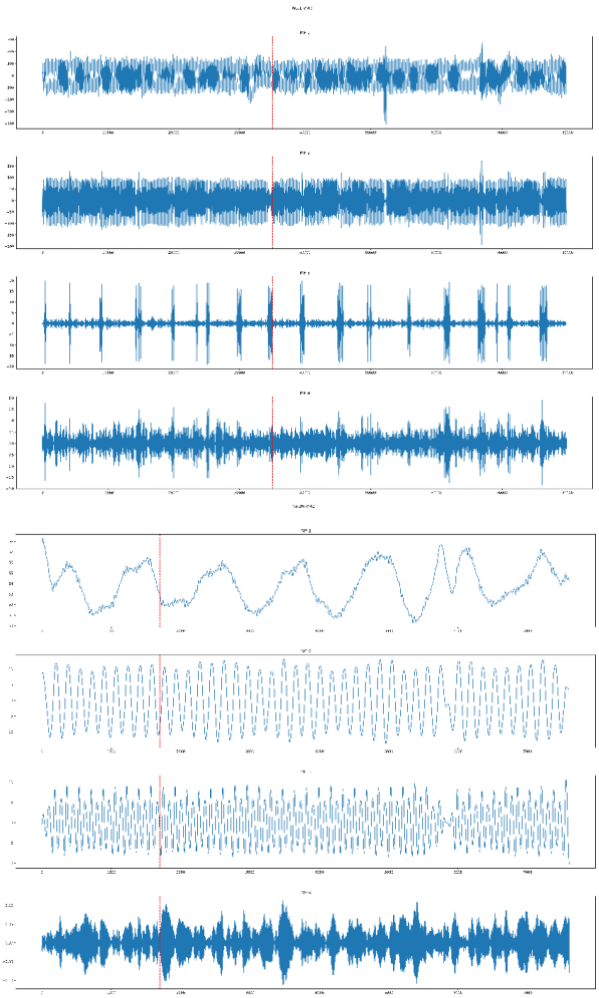


Figure 16: VMD on data set 1 and 26.

We could see that for some data sets, VMD could get a clearer main data sets and help us to analyze the series. We also **try to combine VMD approach with 1-order approach**, but this does not take the same effect as only using VMD approach

### 3.7.4 Pros & Cons of VMD algorithm

It is too time-consuming and is easily to influence by the selection of $k$ and moderate factor $\alpha$, but it could give us a clearer main data sets.

## 3.8 Model bases on Variational Auto Encoder (CNN)

### 3.8.1 Introduction

Variational Auto Encoder is another unsupervised anomaly detection approach. Compared with Auto Encoder, VAE tends to generate data. As long as the decoder is trained well, we can generate data from a standard normal distribution (in an interval) as the input of the decoder to generate similar data, but not exactly the same as the training data. The function of VAE is similar to the role of GAN.

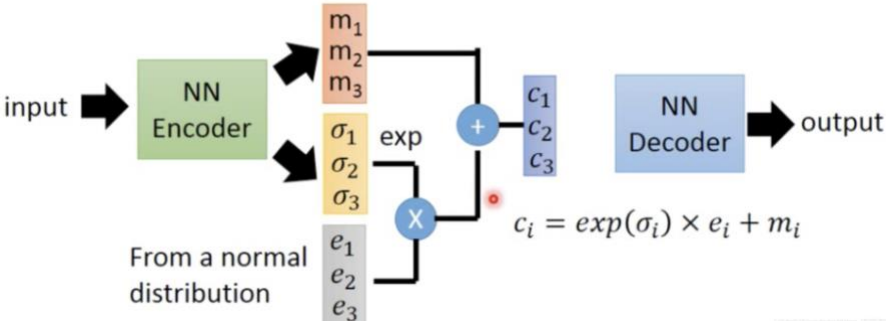



Figure 17: VAE Structure.

In practice, we set the encoder and the decoder as CNN, and we use KL divergence and the sum of Euclidean Distance as the loss.

### 3.8.2 Result

We try 2 approaches to load the data to CNN, and we use Conv1d to scan them. The 2 approaches look like below:

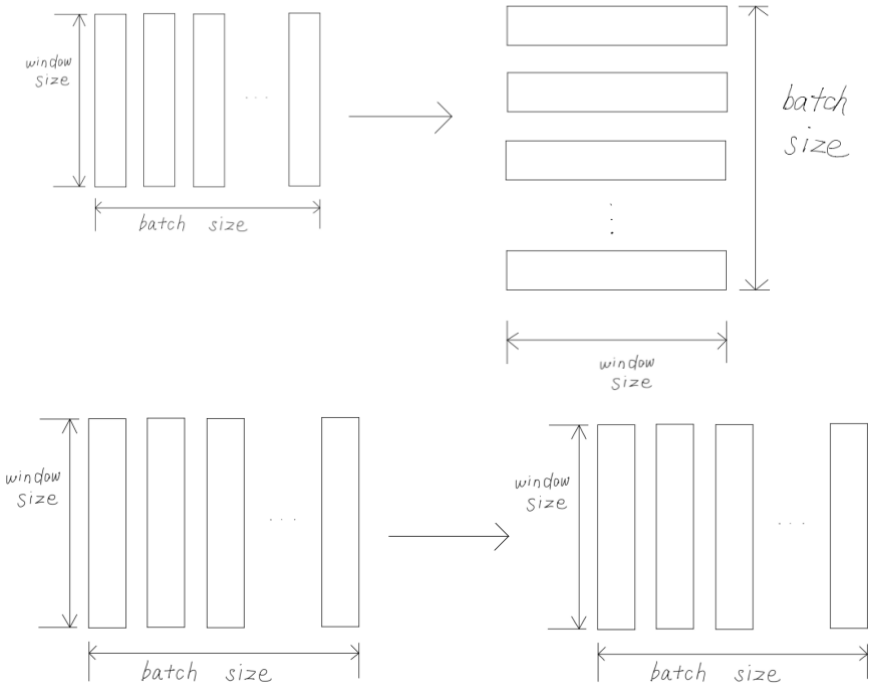



Figure 18:Data loading approaches to CNN.

But actually, for each approach, we train a lot of epochs, when train data is over-fitting, the loss of test data could not decease, and also, the training process is too time-consuming. So, we finally give up this approach. Below are some slices results after some epochs:

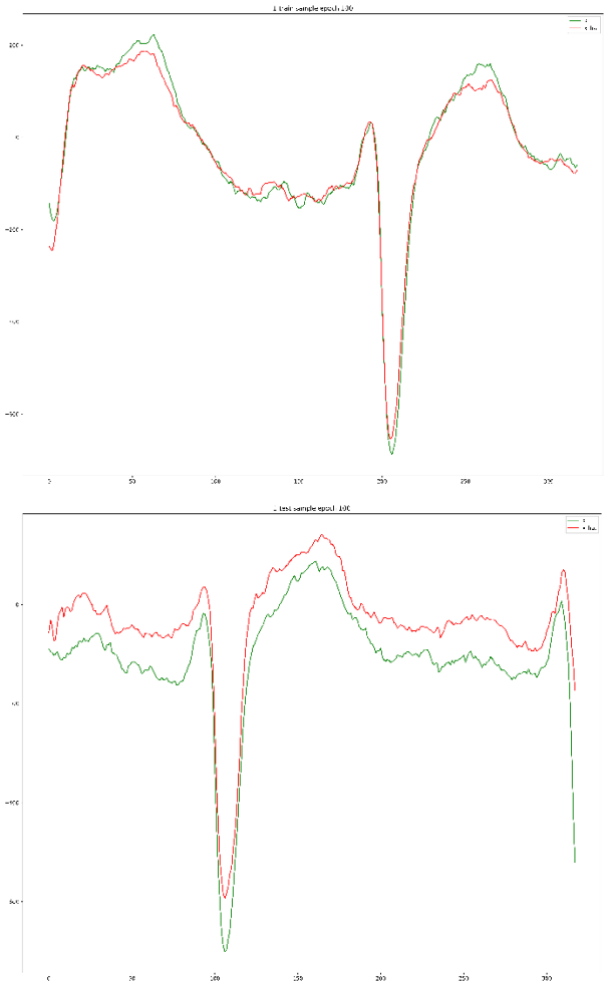
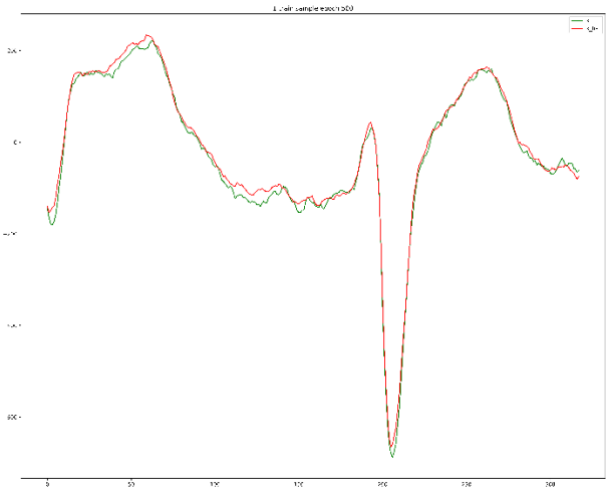


Figure 19: training and test slices after 100 epochs, training loss is 204.1, and test loss is 4909.3.



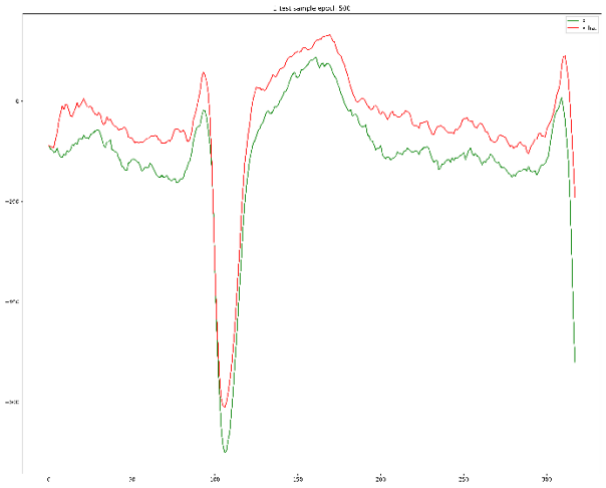


Figure 20: training and test slices after 100 epochs, training loss is 84.0, and test loss is 4953.4.

We could see after 500 epochs, the test loss is still very high, and out calculating and time resource are limited, so finally we give this approach up.

## 3.9 Model based on regression (LSTM)

### 3.9.1 Introduction

We use LSTM model to do the regression task. We use training set to train the model, and use the test set to predict, hoping the anomaly could appear.

### 3.9.2 Result

Actually, the result is not as good as we imagine. Firstly, the time is too consuming, only 5 epochs need more than 5 hours, and this is under the condition that we just put 1 hidden layer. Secondly, the training effect is not so good, both the training and test loss do not decrease obviously, even decrease. Below are some slices after some epochs:

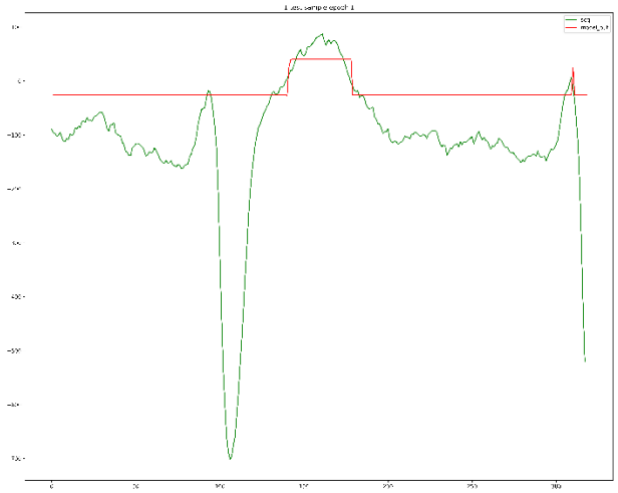


Figure 21: training and test slices after 1 epochs, training loss is 98.6, and test loss is 99.4.

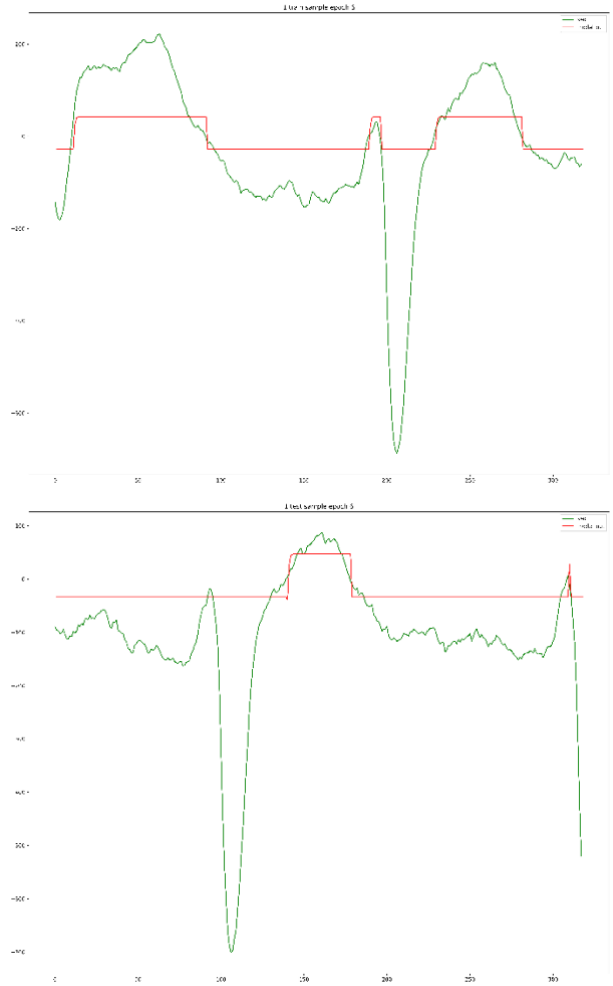


Figure 22: training and test slices after 6 epochs, training loss is 93.2, and test loss is 100.4.

We finally give this approach up due to the high time-consuming and low learning efficiency.

# 4. Ensemble

## 4.1 Model Score

For output of each model, we define the score of each model by the following formula:

$$Score\ of\ model = \frac{highest\ peak\ of\ the\ output}{second\ highest\ peak\ of\ the\ output}$$

Where, we think this formula highlights the effect that the model makes the anomaly stand out. Especially, we get local (30 parts) peak to count it, but not all the test data.

## 4.2 Ensemble approaches

### 4.2.1 Square weighted approach

If the statistical model gets the highest score, we just regard the point standing out in the statistical model as the anomaly center, because the statistical model is more direct and explainable.

Otherwise, we assume that the score of model $k$ is $S_k$, so we re-calculate the score of each model $k$ as $\frac{S_k^2}{\sum_{i=1}^{n} S_i^2}$.

And for the test points in each model, we use standardization and normalization to let each test point get a score in current model. Finally, we sum the scores of test points in each model, and we regard the test point with the highest score is the anomaly center.

### 4.2.2 Dominant approach

We just use the model with the highest score as the predictive model, so regard the most standing-out point in this model as the anomaly center.

## 4.3 Result

We finally choose the Dominant approach as our ensemble approach, because we think the first approach would possibly weaken the contribution of the anomaly point to the final result.

The dominant times for each time are as below:

Statistical: 119

FFT: 36

MP: 32

RRCF: 39

SR: 3

SW: 14

VMD: 7

## 5. Summary

Our approaches include so many models, and this is a long war for us but we enjoy it. We learn so many strategies from this project, although we refer to some others' work on models and codes, but most work are done by ourselves, it is a really great challenge to us.

Our sincere and hearty thanks and appreciations go firstly to our supervisor, Prof. Chen, and all the TAs, whose suggestions and encouragement have given us much insight into these models. It has been a great privilege and joy to study under his guidance and supervision. Furthermore, it is our honor to benefit from his personality and diligence, which we will treasure our whole life.

## 6. Reference

[1] https://github.com/intellygenta
[2] https://www.cnblogs.com/LXP-Never/p/11558302.html#blogTitle6
[3] https://www.linkedin.com/pulse/robust-random-cut-forest-rrcf-math-explanation-logan-wilt/
[4] https://www.sciencedirect.com/topics/engineering/variational-mode-decomposition
[5] https://blog.csdn.net/hahahahah123456/article/details/114080315
[6] *Variational mode decomposition* -Dragomiretskiy.
[7] *Time-Series Anomaly Detection Service at Microsoft* -Hansheng Ren, etc.