LAPPENRANTA UNIVERSITY Of TECHNOLOGY (LUT)

Advanced Data Analysis & Machine Learning (BM20A6100)

Practical Assignment

Fault detection in an industrial process

Submitted By: Anushka Nahid

Student ID: 0532623

Group: F (Individual)

Introduction

According to Penha & Hines (2001), Principle Component Analysis (PCA) could be a data-driven modeling strategy that reconstructs a set of correlated factors into a littler set of modern factors (vital components) that are uncorrelated and hold most of the first data. Hence, a reduced dimension PC demonstrate can be utilized to identify and analyze anomalies within the unique framework in a strong way. Wise et al. (1996) have addressed the use of Principal Component Analysis (PCA) modeling in the monitoring and fault detection of process sensors. In this report, it will explore the use of principal component analysis (PCA) and T^2 measure to detect and distinguish damages in structures. According to Mujica et al (2010), T^2 index is a measure of the variation of each sample within the PCA model. In her opinion, Q-statistic indicates how well each sample conforms to the PCA model. It is a measure of the difference or residual between a sample and its projection into the principal components retained in the model. Principle component analysis (PCA) and Dynamic principle component analysis (DPCA) can detect faults from the data accurately.

Data:

In this practical assignment 21 datasets are provided by the supervisor. I have selected four data sets which are d00_te, d02_te, d09_te & d18_te. Data set d00_te is the original data set, which is in a normal condition, but the other three data set has fault and d09_te is the most difficult one.

Methods:

To detect the fault in the data set, I have used two-dimension reduction techniques which are Principle component analysis and dynamic principle component analysis.

Principle Component Analysis (PCA):

According to Russel, Chiang & Braatz (2000), PCA is an ideal dimensionality diminishment strategy in terms of capturing the fluctuation of the information. PCA decides a set of orthogonal vectors, called loading vectors, which can be requested by the sum of change clarified within the stacking vector bearings. PCA takes advantage of excess data existent in profoundly related factors to diminish the dimensionality. Raich & Cinar (1995) states that PCA decides the most precise lower dimensional representation of the information, in terms of capturing the data directions that have the most variance. The subsequent lower dimensional model has been utilized for distinguishing crazy status and for diagnosing issues leading to the anomalous procedure activity. Kresta et al (1991) & Raich & Cinar (1995) have used PCA for fault diagnosis because of the simplification and the orthogonal property can be obtained from principle component analysis. According to Guo, Li & Laverty (2013) PCA works under three conditions-

- data follows a multivariate normal distribution
- there exists no autocorrelation among observations

- the variables are stationary (keep constant mean and standard deviation over time).

Dynamic Principle Component Analysis (DPCA):

Vanhatalo, Kulachi & Bergquist (2017) states that Dynamic PCA (DPCA) has been suggested as a remedy for high-dimensional and time-dependent data. In DPCA the input matrix is augmented by adding time-lagged values of the variables. In building a DPCA model the analyst needs to decide on (1) the number of lags to add, and (2) given a specific lag structure, how many principal components to retain.

Data matrix of DPCA:

In dynamic PCA, trajectory matrix has been created with time lag. Let matrix X be a set of historical data made up of n observations from p variables, which is taken from a PMU recordings of a distributed power system dynamic process. When the process is under nominal conditions and operates around an operating point, it can be described by the matrix.

$$X = \left[X1\, X2 \ldots\ldots\ldots\ldots X_P \right]_{(n \times p)}$$

Each column in X represents an auto-correlated time series, where a current value depends on the past values. In order to include the serial correlation of data, the so called “trajectory matrix” is constructed applying a time lag shift of order w, on each of the p columns of the matrix X. The trajectory matrix lagged from one variable (Vanhatalo, Kulachi & Bergquist ,2017).

$$X_i^w = \begin{bmatrix} X_i(1) & X_i(2) & \cdots & X_i(w) \\ X_i(2) & X_i(3) & \cdots & X_i(w+1) \\ \vdots & \vdots & \ddots & \vdots \\ X_i(n) & X_i(n+1) & \cdots & X_i(n+w-1) \end{bmatrix}_{(n \times w)}$$

Figure: Trajectory matrix (Vanhatalo, Kulachi & Bergquist ,2017)

Procedure 1:

- At first, I have normalized the data set in the scale of 0 and 1.
- I have used ‘center’ and ‘maverage’ function for smoothing the normalized data set.
- After that I have applied ‘pca’ function in the smoothed data set. He re I used built in matlab function. Hoteling T^2 can be found from this function.

Procedure 2:

- Created a trajectory matrix from the original data set with a function called ‘time_lag_matrix’.
- Normalize the trajectory matrix in the scale of 0 & 1.
- Apply ‘pca function’ to the normalized trajectory matrix.
- Change the lag number in the function and visualize the difference with different lags.

Result Analysis:

The fault in the variables can be explained by visualizing the biplot and by the value of hotelling T^2. In this practical assignment four data set has been used and each data set is compared with the original one. The result will be compared between moving average approach and in time series approach. Testing data set is used in this practical assignment and there is no need of threshold statistics.

Original data set (d00_te) and faulty data set (d02_te):

Original data set is under normal condition. PCA has been applied to moving average approach and in time series approach. In time series approach, lag operator has been used. The time is shifted by lag 1,2,3 and 6. When the time is shifted, the column number is increasing. When data is shifted by lag 1, data with seasonality will repeat itself periodically in a sine or cosine wave. Data is shifted the time series by 1. So, in lag operation when the data is shifted by time 1, each column is shifted by time 1. That’s why column number increased and goes from 52 to 104.

When moving average and PCA are applied to original data set, the fault behaviour causes for variable number 50, 18, 19,11,35,47,33,7, 13, 16, 31,46,20. When the data set is shifted by lag 1, the fault behaviour is also start from same variable number as in moving average. When the data is shifted by lag 2,3 and 6, the fault behaviour in variables are same. If the fault is started in 18th variable in lag 1, the fault causes in 36th variable in lag 2. These faults are found by visualizing biplot. The variables which have the highest length, I considered them as fault.

In data set 2(d02_te), the variables which causes fault in moving approach, they are- 34,28,19,50,11,22,35,47,10,18,3,43,30. When DPCA has been done in data set 2, the fault behaviour has been started in same variables as in moving average approach but two variables are added in this approach which are 46 and 36. If there is fault in 34th variable in lag operator 1, the fault has been shifted in 102th variable in lag operator 3.So time is shifted by 3; like 34*3=102.

When I have plotted the figure of each column in original data set and in data set 2, I observed that in some points the fault is starting and after sometimes it goes back to normal. Sometimes fault is occurring, and it never goes back to normal. If we are comparing the time variant graph of original data set and data set 2, the fault in 28th variable starts from observation 210 and it never goes back to normal. There are faults in 34th ,19th, 50th, 22th ,35th, 47th ,10th ,3rd ,43th , 30th variables and the fault behavior continues and it never goes back to normal situation. There are faults in other variables but after a certain observation the faults behavior ends, and it goes back in normal condition.

Original data set (d00_te) and difficult data set 9 (d09_te):

When moving average and PCA are applied to original data set, the fault behaviour causes for variable number 50, 18, 19,11,35,47,33,7, 13, 16, 31,46,20. When the data set is shifted by lag 1, the fault behaviour is also start from same variable number as in moving average. When the data is shifted by lag 2,3 and 6, the fault behaviour in variables are same. If the fault is started in 19th variable in lag 1, the fault causes in 114th variable in lag 6.

In difficult data set, the variables which causes fault in moving average approach they are 11th, 33rd, 38th ,13th ,7th ,46th ,16th, 31st variable. When trajectory matrix is created from original matrix and data is shifted by time 1, the fault behaviour occurs in variables 33rd, 11th ,38th ,13th, 7th ,46th, 16th ,31st. There are some variables which has fault in time series matrix and are different from moving approach, they are 44th, 1st, 27th variables. When time is shifted by lag 1, the fault occurs in 11th variable and when the time is shifted by lag 6 the fault shifted from 13th to 78th variable. As this is the most difficult data set, it was very hard to find from which observation the fault starts and where it is going to stop. In 13th variable, the fault starts from 80th observations and stays till 100th observation. Then it goes to normal situation again the fault starts from 117th variable and ends at 141st variable after that it goes back to normal but after certain again there is fault. At the end, the fault behaviour becomes normal. 7th, 13th, 16th, 46th ,38th variables cause most difficult fault. In every observation they have asymmetric fault.

Original data set (d00_te) and data set 18 (d18_te):

When moving average and PCA are applied to original data set, the fault behaviour causes for variable number 50, 18, 19,11,35,47,33,7, 13, 16, 31,46,20. When the data set is shifted by lag 1, the fault behaviour is also start from same variable number as in moving average. When the data is shifted by lag 2,3 and 6, the fault behaviour in variables are same. If the fault is started in 31st variable in lag 1, the fault causes in 62nd variable in lag 2.

After applying moving average and PCA in data set 18, the variables who has fault has been found. Variables 33th, 35th, 47th, 37th ,30th, 11th, 24th ,16th, 7th, 13th, 29th , 20th , 23th , 45th ,31th have faults in moving average approach. After doing DPCA in original data set, the fault has been found in variables 33rd, 27th, 35th, 11th ,30th ,47th, 24th, 38th ,13th,7th,16th,20th,45th,29th ,25th. It can be seen that variables 25th ,45th ,24th have been added to fault in DPCA method. In 33rd variable, the fault behaviour starts from 253 observation and it never comes back to normal position.

Hotelling T^2 has been computed in both cases. It is calculating the distance of variables from the centre. Explained is giving the the percentage of the total variance explained by each principal component.

Conclusion:

The Tennessee Eastman process simulator has been used to compare between PCA and DPCA for detecting the faults. It can be concluded that the variables which are causing faults, are same in moving average and in time series process. As the time is shifting so the observation number is reducing also. If any threshold is applied for detecting the fault, the result will be more accurate.

References:

Guo, Y., Li, K., & Laverty, D. M. (2014). Loss-of-Main Monitoring and Detection for Distributed Generations Using Dynamic Principal Component Analysis. Journal of Power and Energy Engineering, 2(4), 423-431.https://doi.org/10.4236/jpee.2014.24057.

Penha, R.M.L., Hines J.W., (2001), Using Principal Component Analysis Modeling to Monitor Temperature Sensors in a Nuclear Research Reactor. SP 05508-900, TN 37996-0750.

Vanhatalo, E., Kulahci, M., & Bergquist, B. (2017). On the structure of dynamic principal component analysis used in statistical process monitoring. Chemometrics and Intelligent Laboratory Systems, 167, 1–11. doi:10.1016/j.chemolab.2017.05.016

Burgos, D.A.T. and Mujica, L.E. and Guemes A. and Rodellar J. (2010) Active piezoelectric system using PCA . In European Workshop on Structural Health Monitoring. 5th edition. Sorrento-Italy.

Raich, A. C., & Çinar, A. (1995). Multivariate statistical methods for monitoring continuous processes: assessment of discrimination power of disturbance models and diagnosis of multiple disturbances. Chemometrics and Intelligent Laboratory Systems, 30(1), 37–48. doi:10.1016/0169-7439(95)00035-6

Russell, E. L., Chiang, L. H., & Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems, 51(1), 81–93. doi:10.1016/s0169-7439(00)00058-7

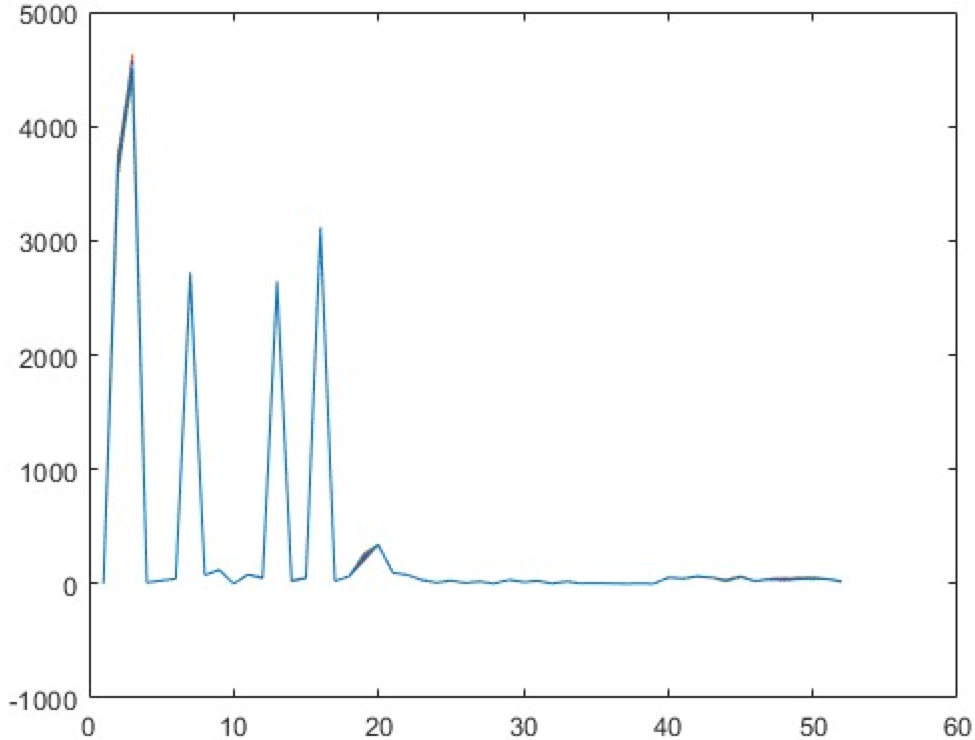
Kresta, J. V., J. F. MacGregor and T. E. Marlin (1991). Multivariate Statistical Monitoring of Process Operating Performance. The Canadian Journal of Chemical Engineering 69, 35—47.

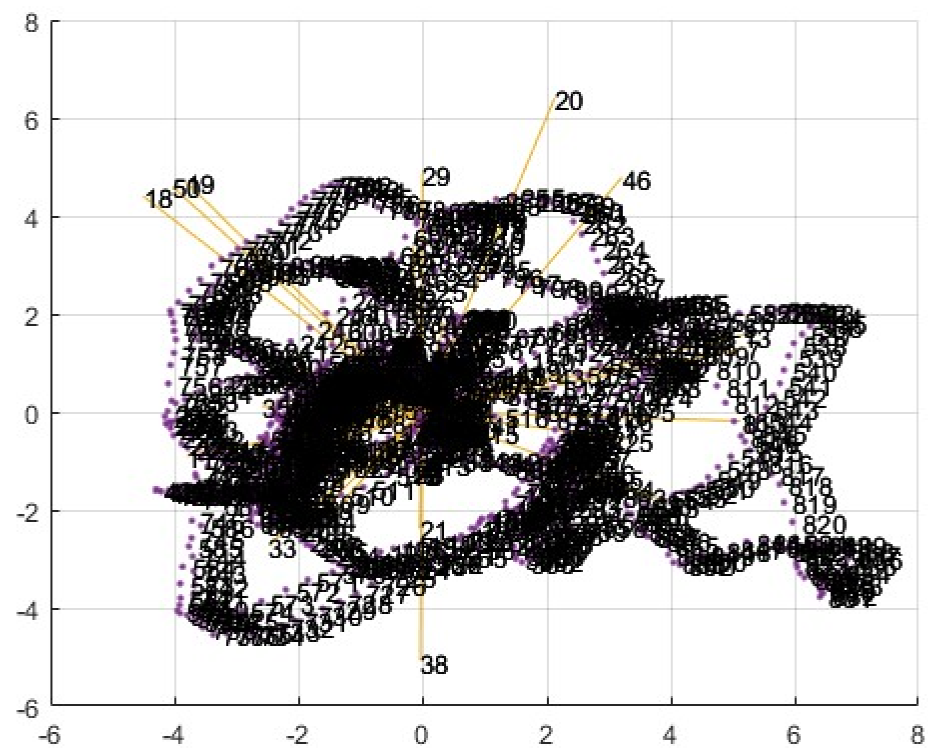
Wise, B.M. and Gallagher, N.B. (1996) The Process Chemometrics Approach to Process Monitoring and Fault Detection. Journal of Process Control, 6, 329-348. http://dx.doi.org/10.1016/0959-1524(96)00009-1

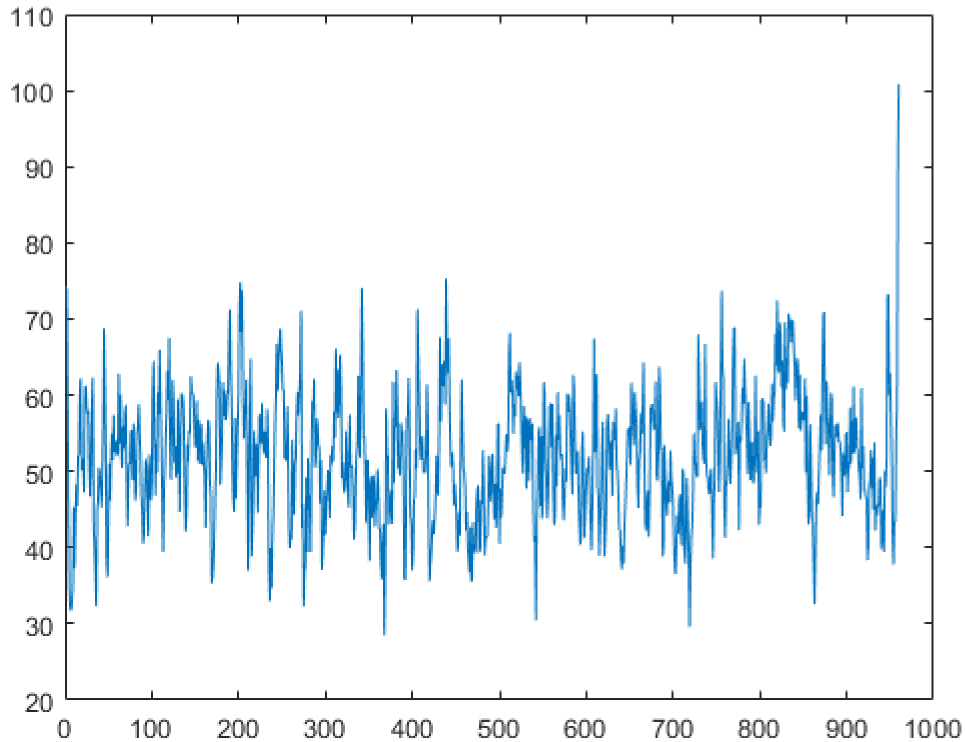
ST0532623: case:1 - original data set (no fault)

```
clc
clear all

load ('d00_te.dat') % load the data set
figure(1)
plot(1:52,d00_te) % plot the original data set

% We have to normalize the data set . Here i have used matlab built in
% function 'normalize'. It will scale the original data in between 0 & 1.
normalize_data=normalize(d00_te);

% After normalizing the data I will use 'center' & 'maverage' function from
% the 'datana' toolbox for centering the data and smotthing the data.
data_centered_original = center(maverage((normalize_data),13,0,1));

% Now i have applied matlab builtin function 'pca' to the center data set
% in this function coeff will give the principle component coefficients
% which is known as loadings, Score is the principle component scores,
% latent will return eigenvalues of covariance matrix of the dataset,
% tsquared is known as hotteling tsquare and explained will give the
% percentage of total variance explained by the each principle component.

[coeff0,score0,latent0,tsquared0,explained0]=pca((data_centered_original));
% now i will plot a biplot with scores and loadings of principle componet
% analysis
figure (2); hold on; grid on;
plotbi(score0,coeff0,1,2,1:size(data_centered_original,1),[],1:size(score0,1),1:size(score0,2));
figure(3); plot(tsquared0)
figure(4);plot(explained0)

%  we can determine coefficient of determination from 'explained' output of pca
rsquared=cumsum(explained0);

%DPCA (Dynamic Principle Component Analysis)
% Another apporach (related to Russel article)
% I will create the new matrix. for this i have created a function which name is 'time_lag_matrix'.
% At first i have Created the new matrix with lag 1
```

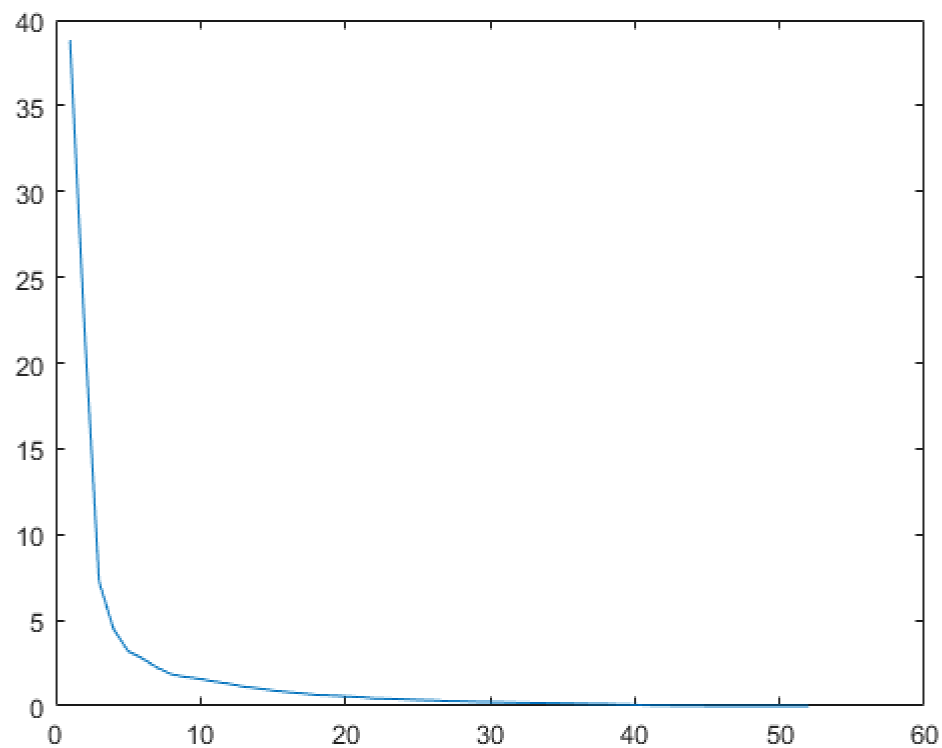
```
lag_mat=time_lag_matrix(d00_te,1);
% After creating the new matrix i have normalized the data in the scale of 0 & 1.
norm_lag_mat=normalize(lag_mat);
% In the new normalize matrix i have applied built in function 'pca'.

[coeff0_new,score0_new,latent0_new,tsquared0_new,explained0_new]=pca((norm_lag_mat));
figure (5); hold on; grid on;
plotbi(score0_new,coeff0_new,1,2,1:size(norm_lag_mat,1),[],1:size(score0_new,1),1:size(score0_new,2));
% plot the hotteling tsquare and explained variables
figure(6); plot(tsquared0_new)
figure(7);plot(explained0_new)
% we can find the new coefficient of determination
rsquarednew=cumsum(explained0_new);

% Now Create the new matrix with lag 2
lag2_mat=time_lag_matrix(d00_te,2);
% normalize the new data
norm_lag2_mat=normalize(lag2_mat);
[coeff02_new,score02_new,latent02_new,tsquared02_new,explained02_new]=pca((norm_lag2_mat));
figure (8); hold on; grid on;
plotbi(score02_new,coeff02_new,1,2,1:size(norm_lag2_mat,1),[],1:size(score02_new,1),1:size(score02_new,2));
figure(9); plot(tsquared02_new)
figure(10);plot(explained02_new)

% Create the new matrix with lag 3
lag3_mat=time_lag_matrix(d00_te,3);
% normalize the new data
norm_lag3_mat=normalize(lag3_mat);
[coeff03_new,score03_new,latent03_new,tsquared03_new,explained03_new]=pca((norm_lag3_mat));
figure (11); hold on; grid on;
plotbi(score03_new,coeff03_new,1,2,1:size(norm_lag3_mat,1),[],1:size(score03_new,1),1:size(score03_new,2));
figure(12); plot(tsquared03_new)
figure(13);plot(explained03_new)

% Create the new matrix with lag 6

lag6_mat=time_lag_matrix(d00_te,6);
% normalize the new data
norm_lag6_mat=normalize(lag6_mat);
[coeff06_new,score06_new,latent06_new,tsquared06_new,explained06_new]=pca((norm_lag6_mat));
figure (14); hold on; grid on;
plotbi(score06_new,coeff06_new,1,2,1:size(norm_lag6_mat,1),[],1:size(score06_new,1),1:size(score06_new,2));
figure(15); plot(tsquared06_new)
figure(16);plot(explained06_new)
```


5000
4000
3000
2000
1000
0
-1000
0
10
20
30
40
50
60

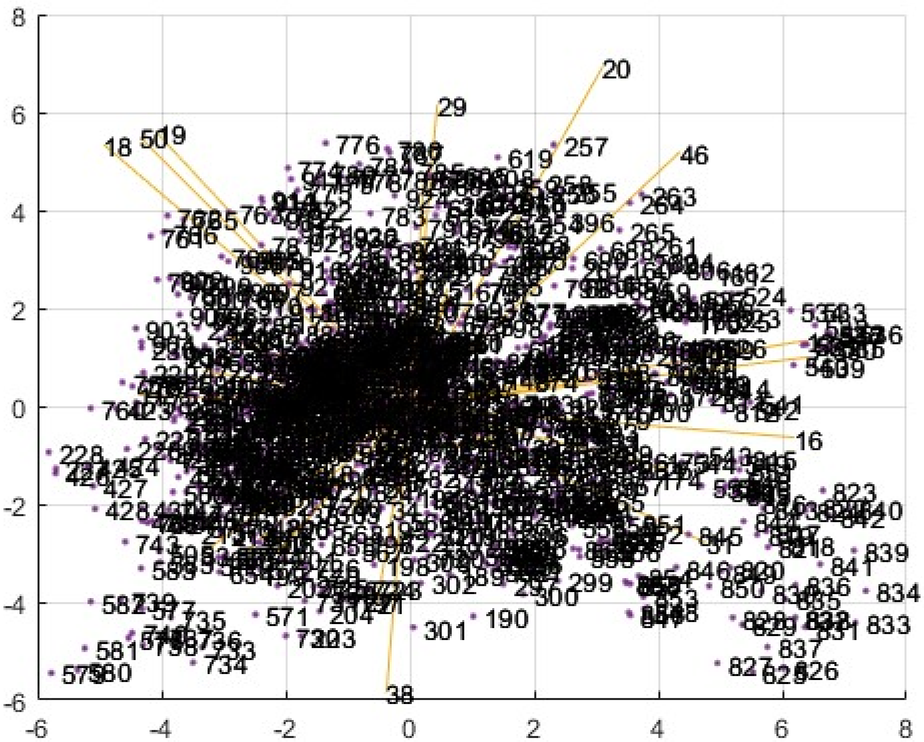

8
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
8
20
29
46
18
50
19
38
21
33
820
819
810

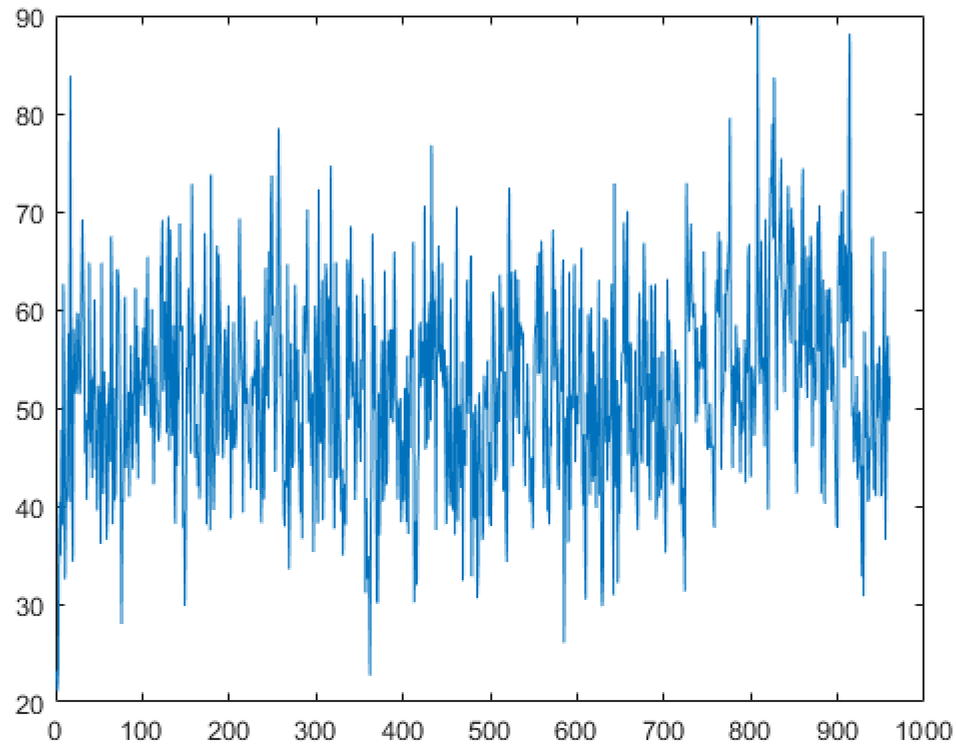


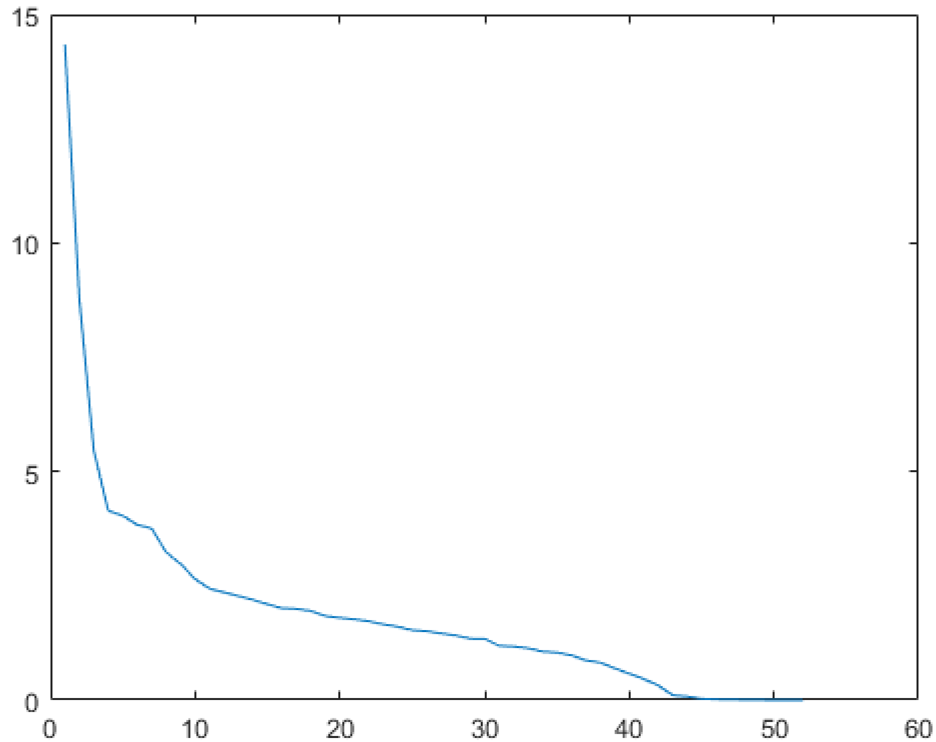
110
100
90
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
800
900
1000

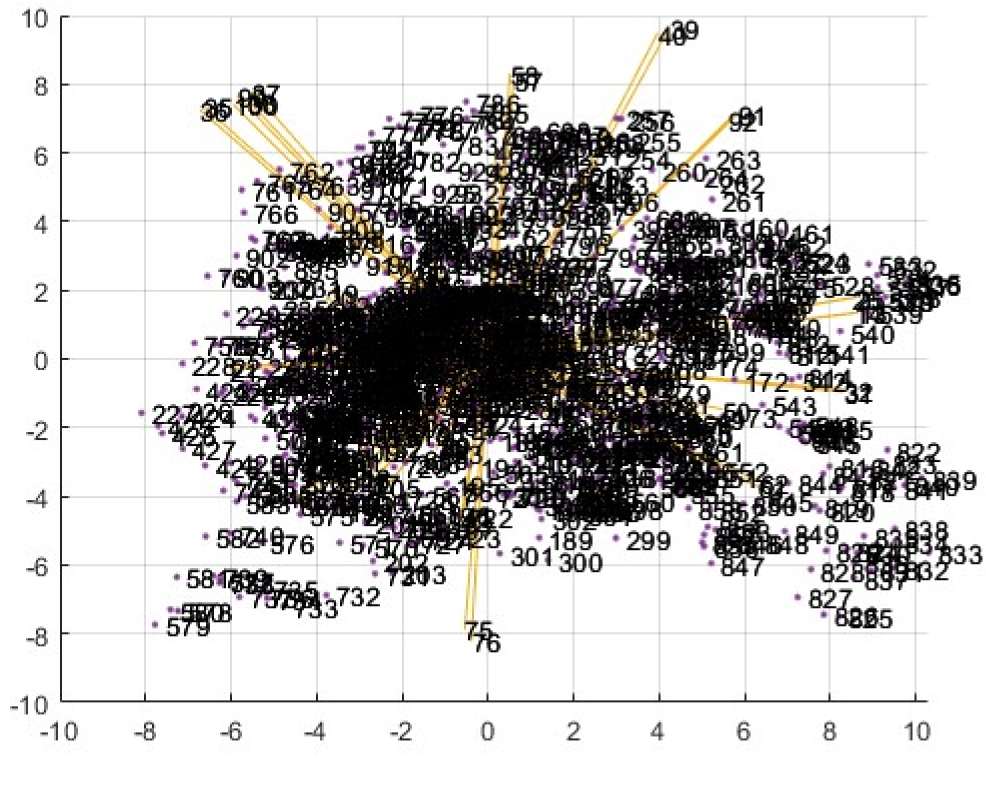


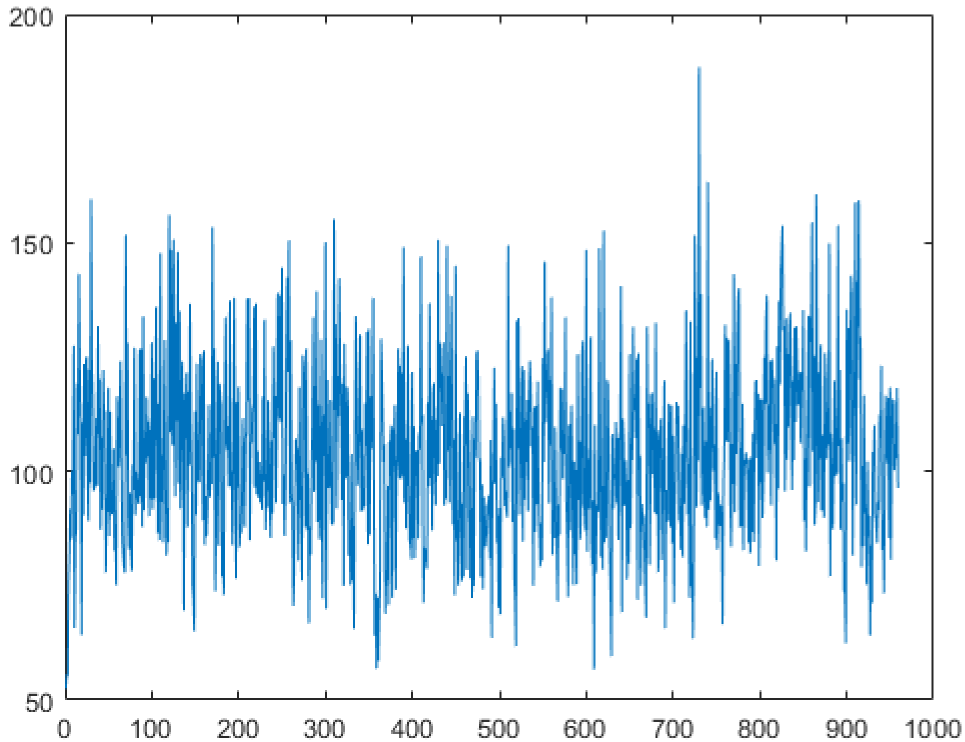
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60

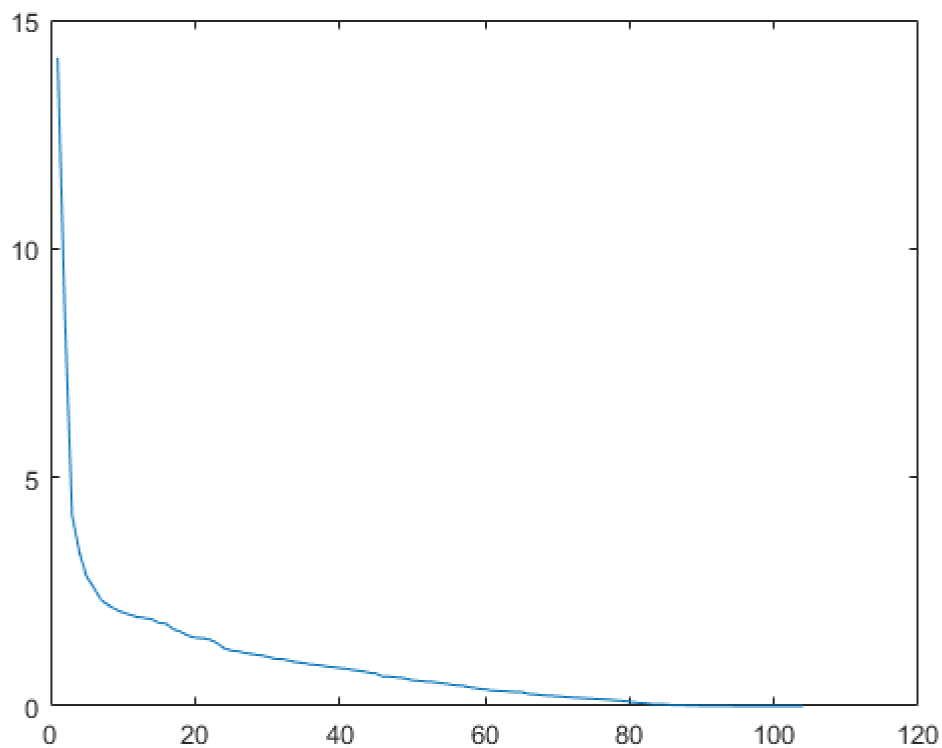

8
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
8

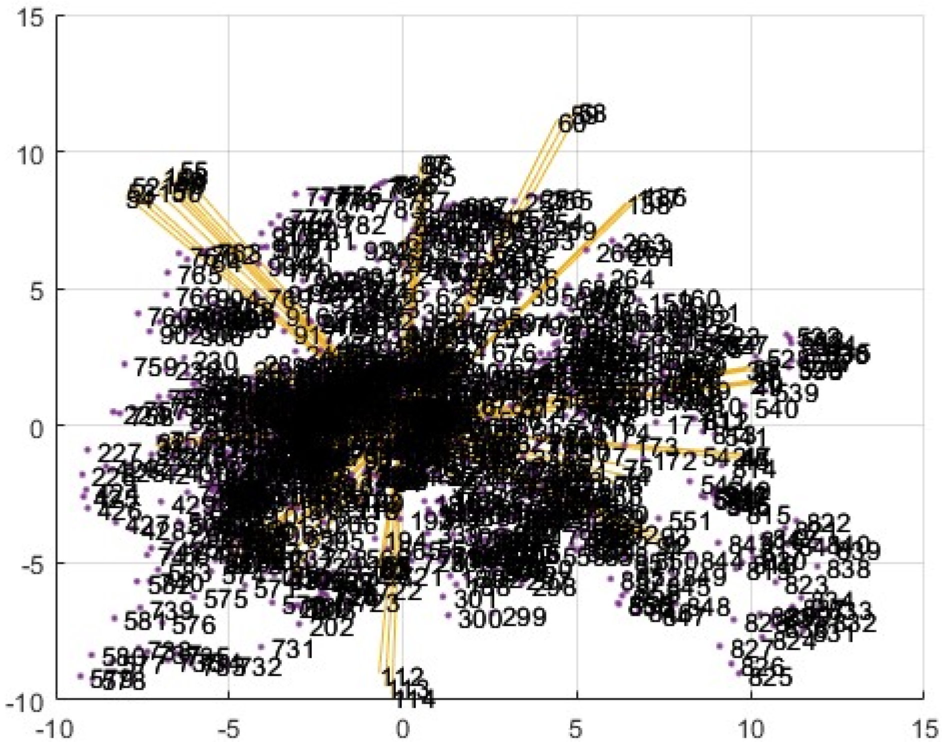

90
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
800
900
1000

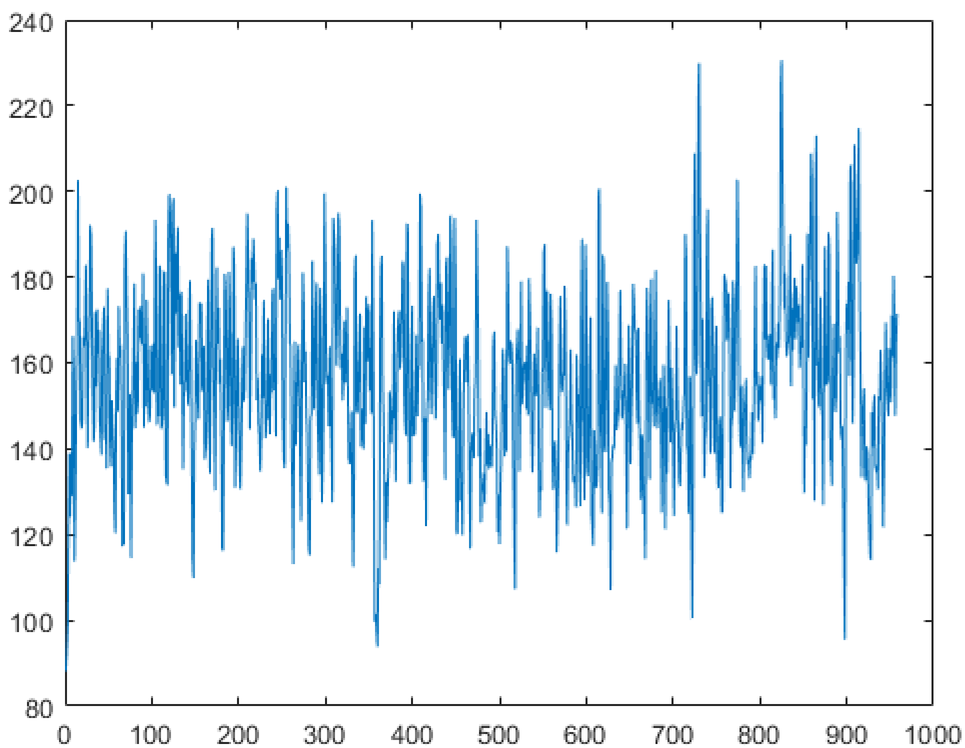

15
10
5
0
0
10
20
30
40
50
60

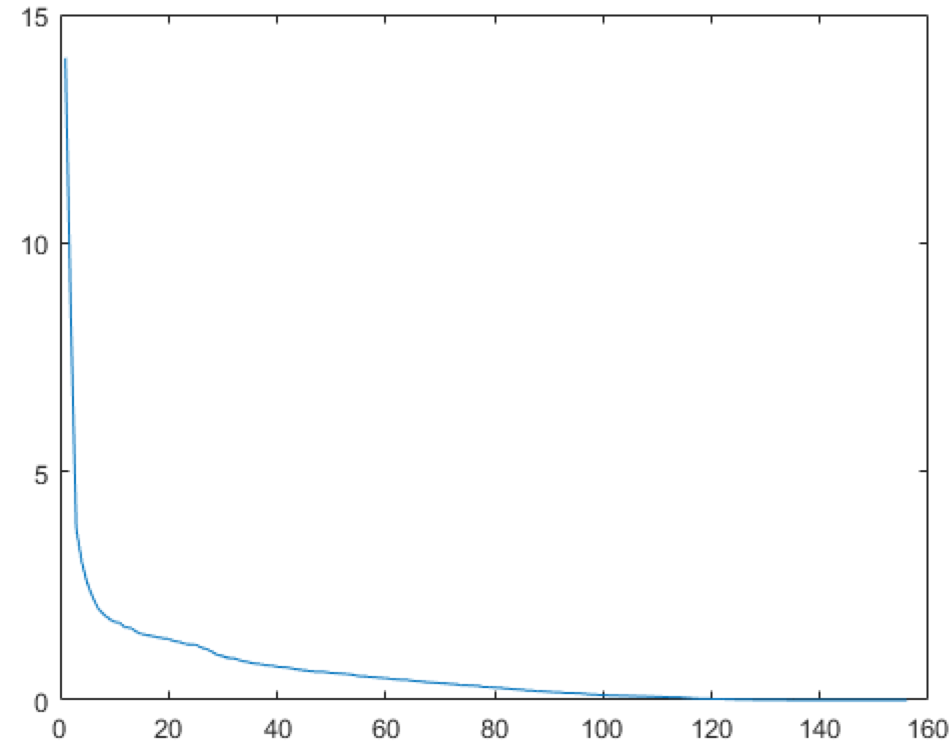

10
8
6
4
2
0
-2
-4
-6
-8
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10


200
150
100
50
0
100
200
300
400
500
600
700
800
900
1000

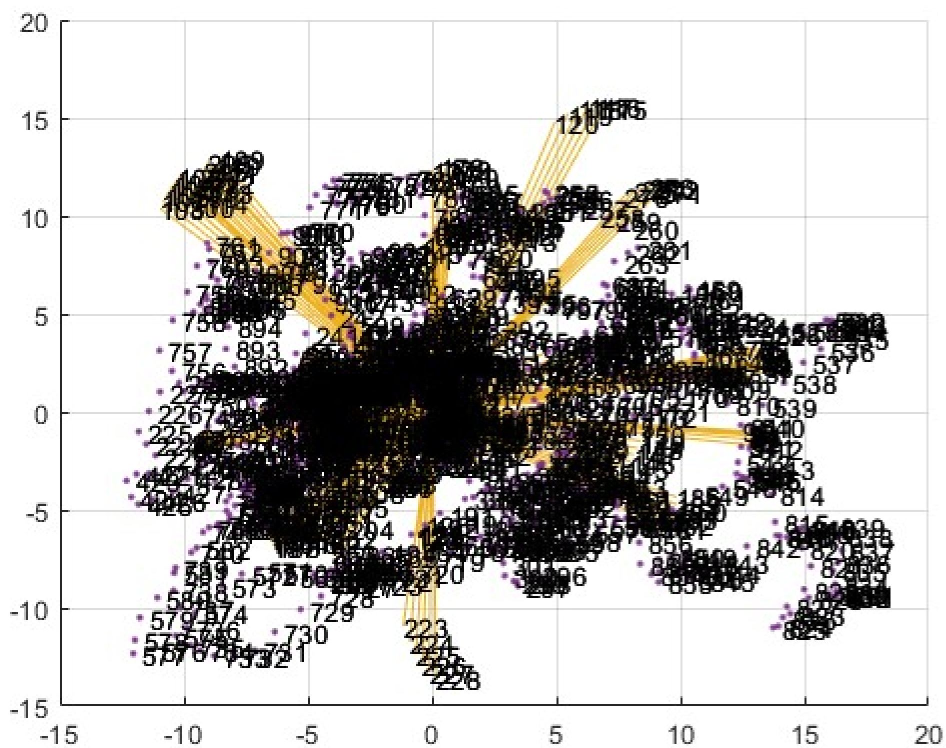

15
10
5
0
0
20
40
60
80
100
120


15
10
5
0
-5
-10
-10
-5
0
5
10
15

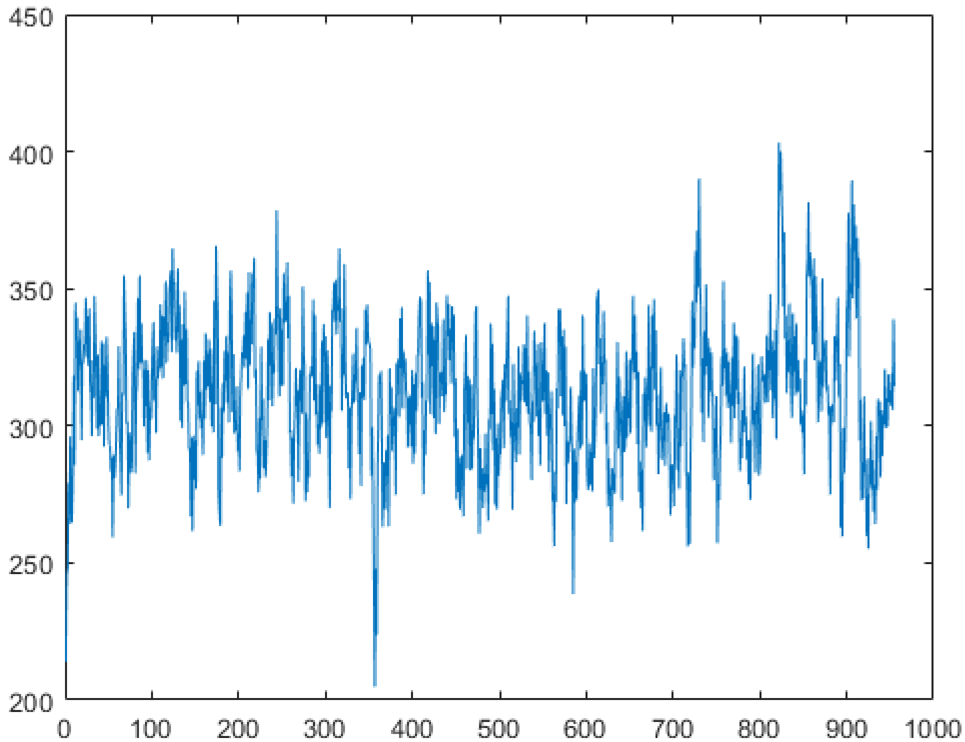

240
220
200
180
160
140
120
100
80
0
100
200
300
400
500
600
700
800
900
1000

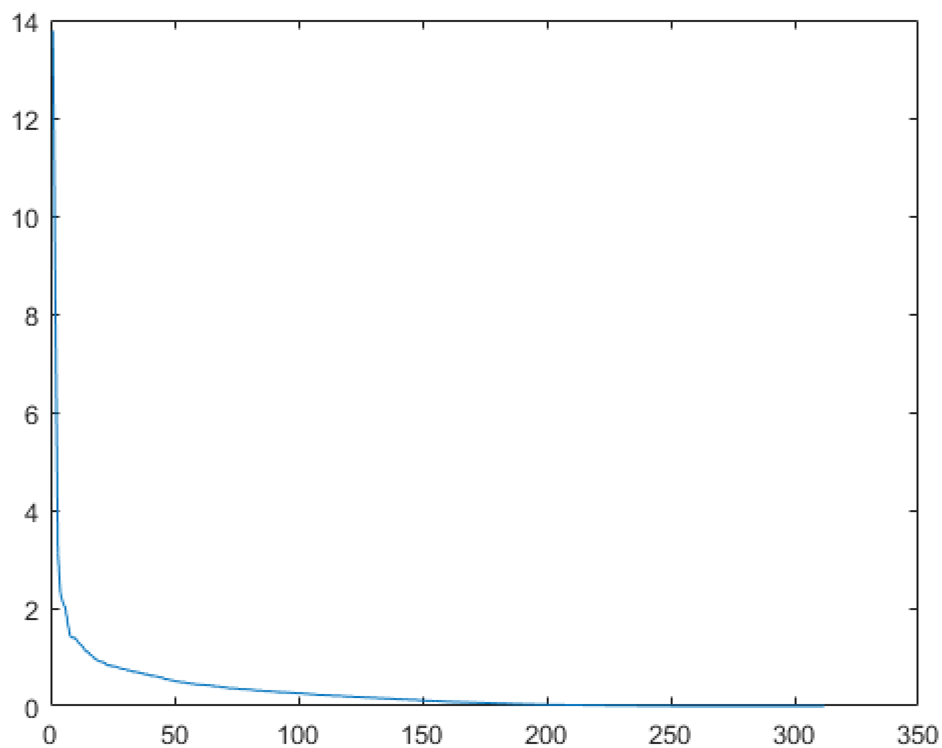


15
10
5
0
0
20
40
60
80
100
120
140
160



20
15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15
20


450
400
350
300
250
200
0 100 200 300 400 500 600 700 800 900 1000

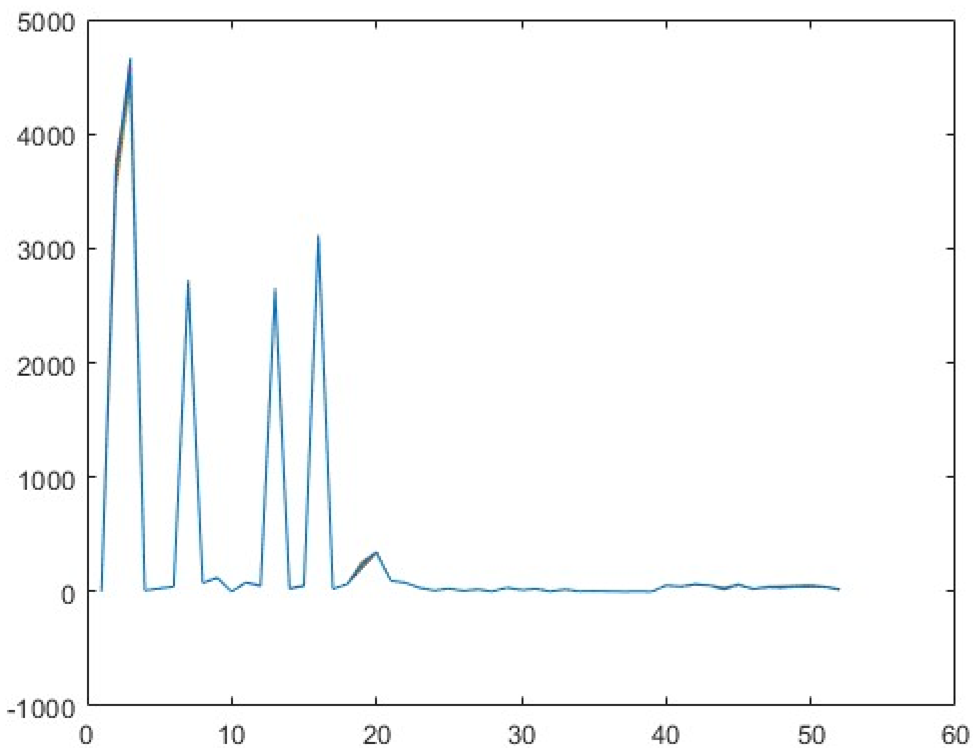

14
12
10
8
6
4
2
0
0 50 100 150 200 250 300 350

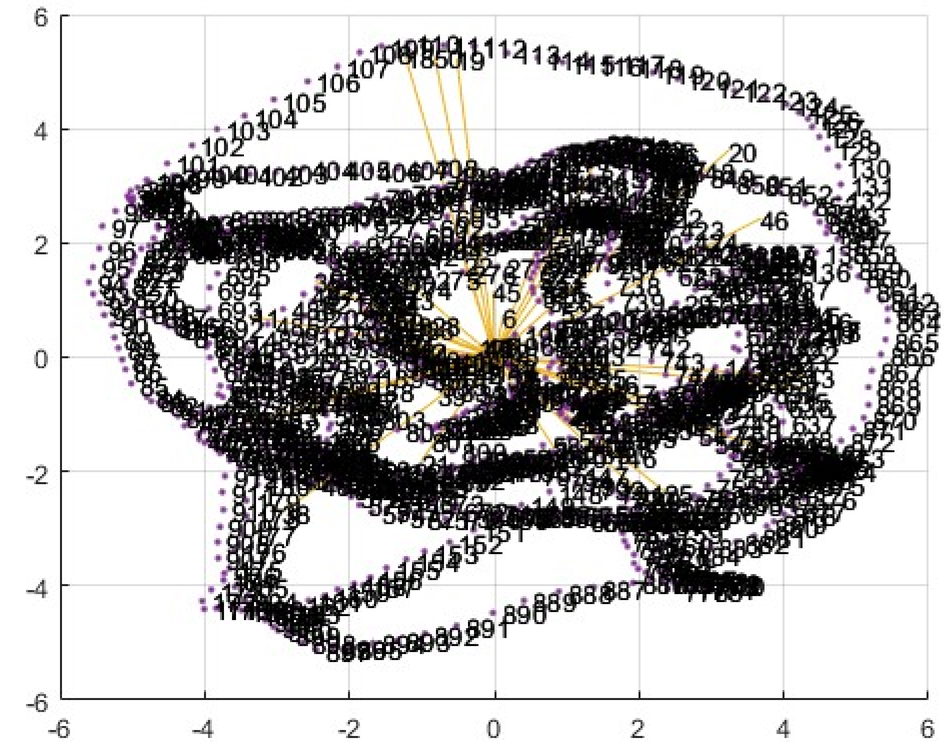
ST0532623: Case:2 (difficult data set: data set 9)

load the data set all the procedure will be same as previous. that's why I ain't comment all the lines as it is repetation.

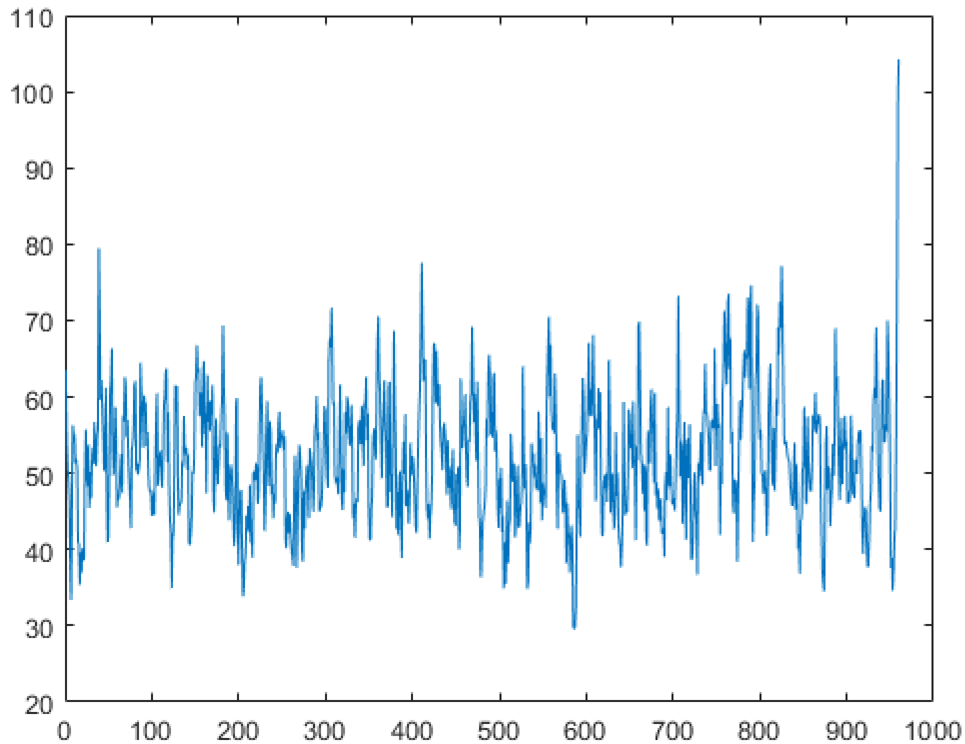
```
load ('d09_te.dat')
figure(17)
plot(1:52,d09_te)
% scale the data set
normalize_data_9=normalize(d09_te);
data_centered_original_9 = center(maverage((normalize_data_9),13,0,1));
[coeff9,score9,latent9,tsquared9,explained9]=pca((data_centered_original_9)); % done pca
figure (18); hold on; grid on;
plotbi(score9,coeff9,1,2,1:size(data_centered_original_9,1),[],1:size(score9,1),1:size(score9,2));
figure(19); plot(tsquared9)
figure(20);plot(explained9)
% coefficient of determination
rsquared9=cumsum(explained9);

%DPCA
% Create the new matrix with lag 1
lag_mat9=time_lag_matrix(d09_te,1);
% normalize the new data
norm_lag_mat_9=normalize(lag_mat9);
[coeff9_new,score9_new,latent9_new,tsquared9_new,explained9_new]=pca((norm_lag_mat_9));
figure (21); hold on; grid on;
plotbi(score9_new,coeff9_new,1,2,1:size(norm_lag_mat_9,1),[],1:size(score9_new,1),1:size(score9_new,2));
figure(22); plot(tsquared9_new)
figure(23);plot(explained9_new)
rsquarednew9=cumsum(explained9_new);

% Create the new matrix with lag 2

lag2_mat9=time_lag_matrix(d09_te,2);
% normalize the new data
norm_lag2_mat_9=normalize(lag2_mat9);
[coeff92_new,score92_new,latent92_new,tsquared92_new,explained92_new]=pca((norm_lag2_mat_9));
figure (24); hold on; grid on;
plotbi(score92_new,coeff92_new,1,2,1:size(norm_lag2_mat_9,1),[],1:size(score92_new,1),1:size(score92_new,2));
figure(25); plot(tsquared92_new)
figure(26);plot(explained92_new)
% rsquarednew9=cumsum(explained92_new)

% Create the new matrix with lag 3

lag3_mat9=time_lag_matrix(d09_te,3);
```

```
% normalize the new data
norm_lag3_mat_9=normalize(lag3_mat9);
[coeff93_new,score93_new,latent93_new,tsquared93_new,explained93_new]=pca((norm_lag3_mat_9));
figure (27); hold on; grid on;
plotbi(score93_new,coeff93_new,1,2,1:size(norm_lag3_mat_9,1),[],1:size(score93_new,1),1:size(score93_new,2));
figure(28); plot(tsquared93_new)
figure(29);plot(explained93_new)
% rsquarednew9=cumsum(explained93_new)

% Create the new matrix with lag 6

lag6_mat9=time_lag_matrix(d09_te,6);
% normalize the new data
norm_lag6_mat_9=normalize(lag6_mat9);
[coeff96_new,score96_new,latent96_new,tsquared96_new,explained96_new]=pca((norm_lag6_mat_9));
figure (30); hold on; grid on;
plotbi(score96_new,coeff96_new,1,2,1:size(norm_lag6_mat_9,1),[],1:size(score96_new,1),1:size(score96_new,2));
figure(31); plot(tsquared96_new)
figure(32);plot(explained96_new)
% rsquarednew9=cumsum(explained96_new)
```

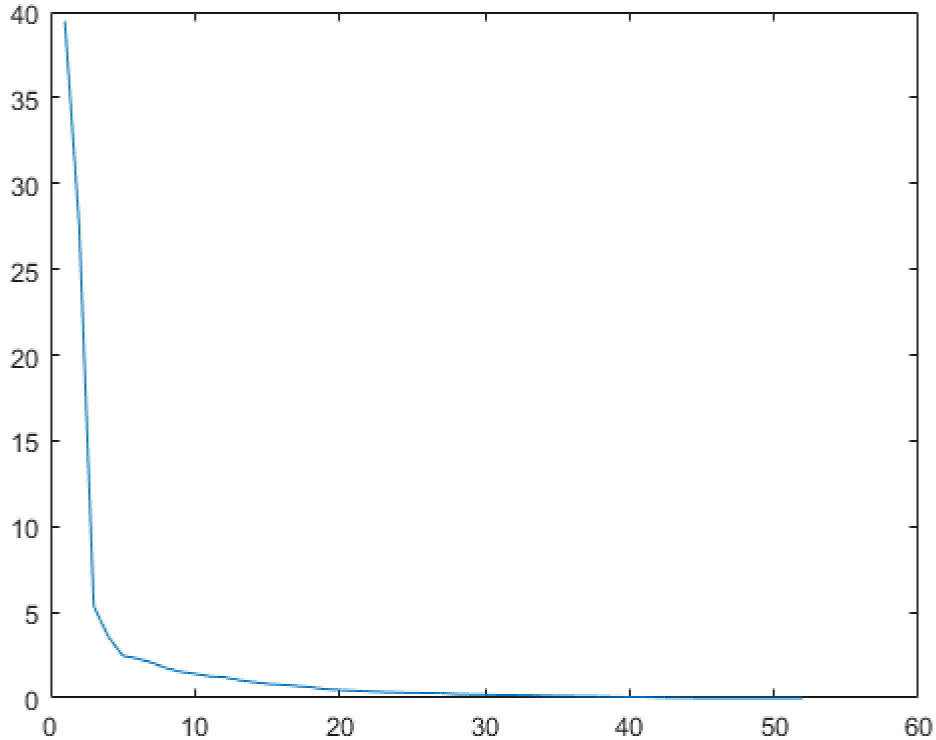




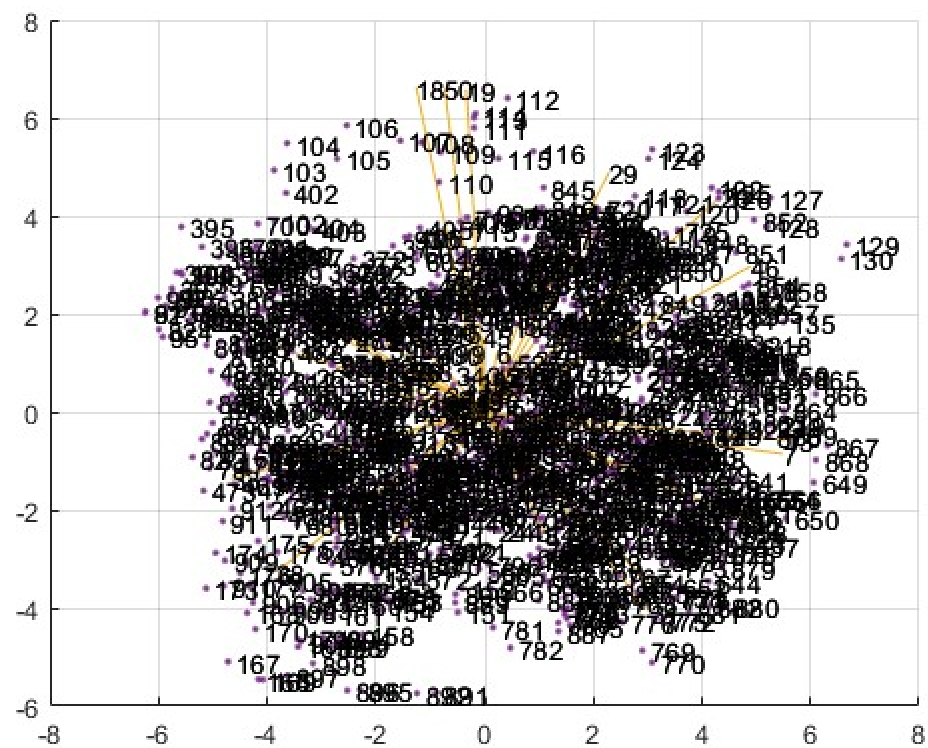
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6



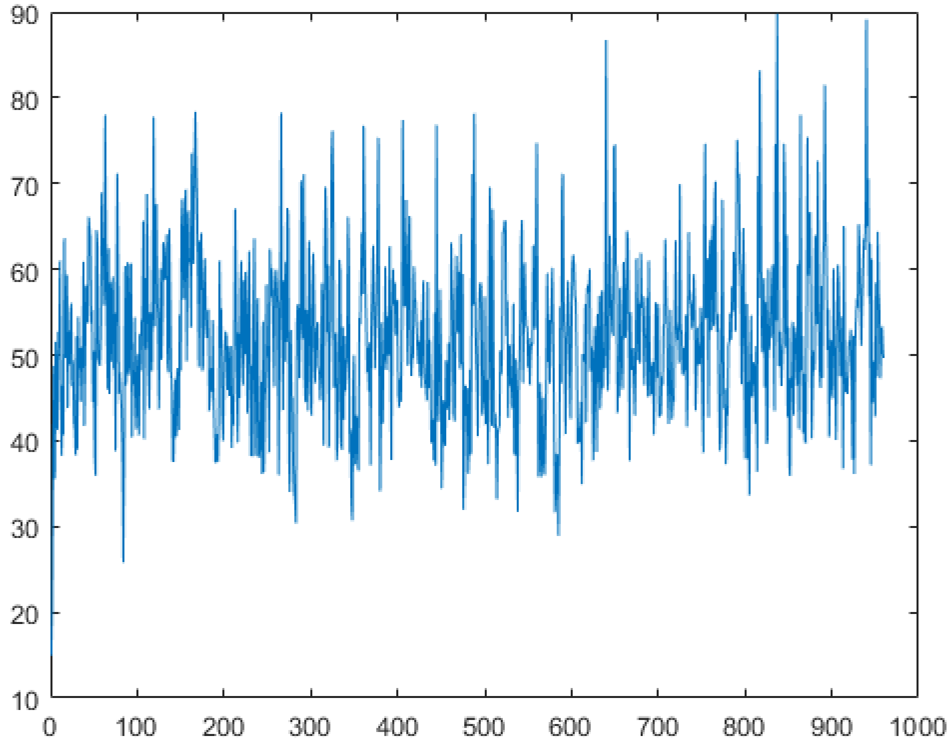
110
100
90
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
800
900
1000

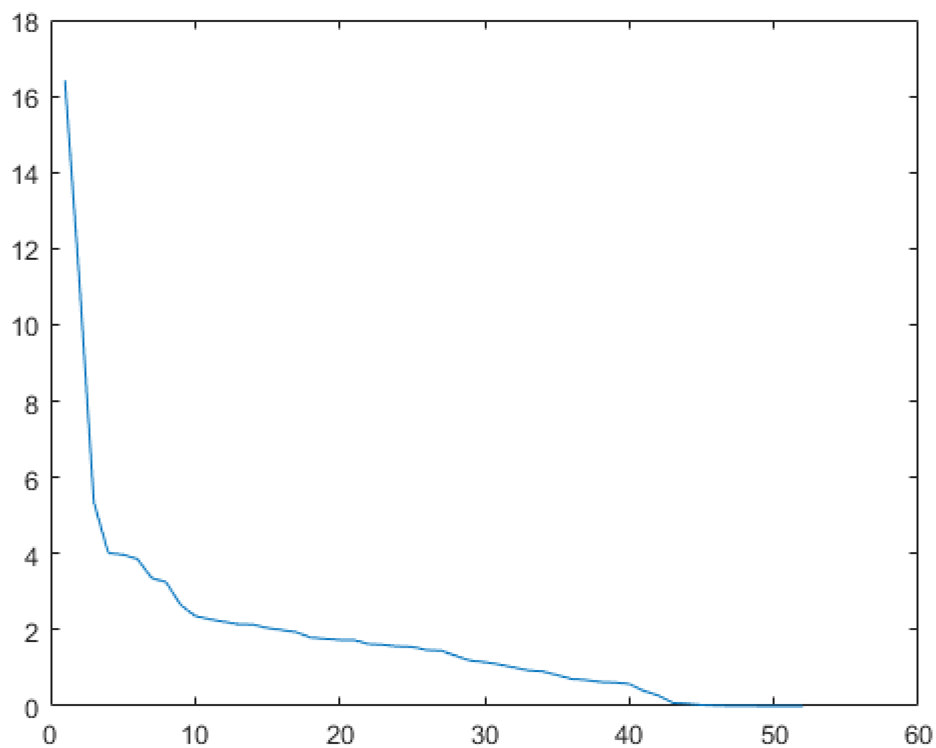


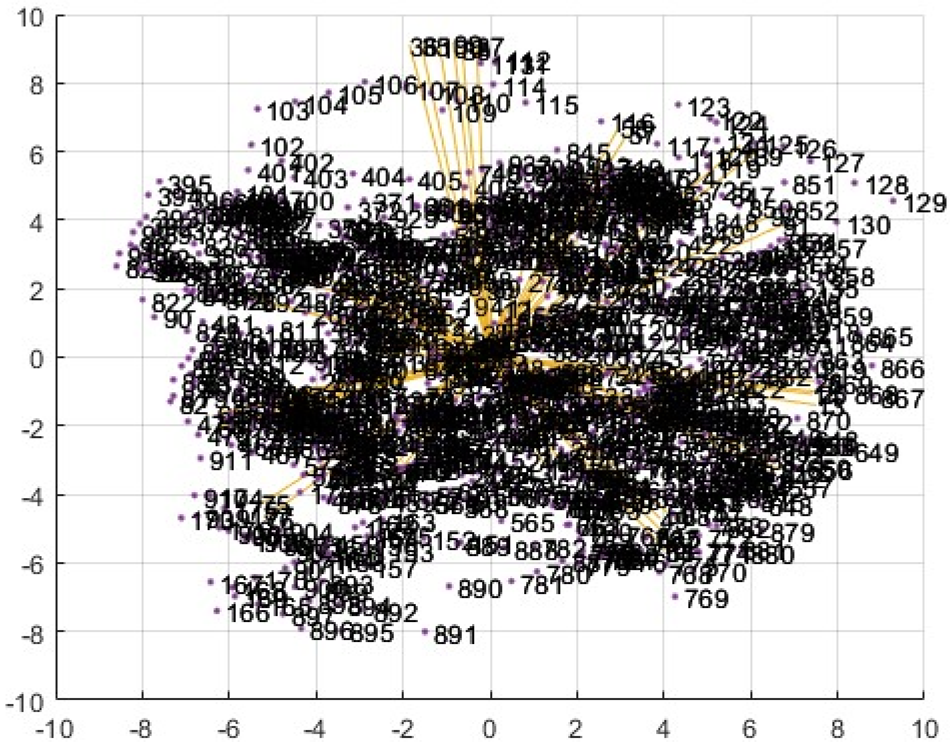
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60

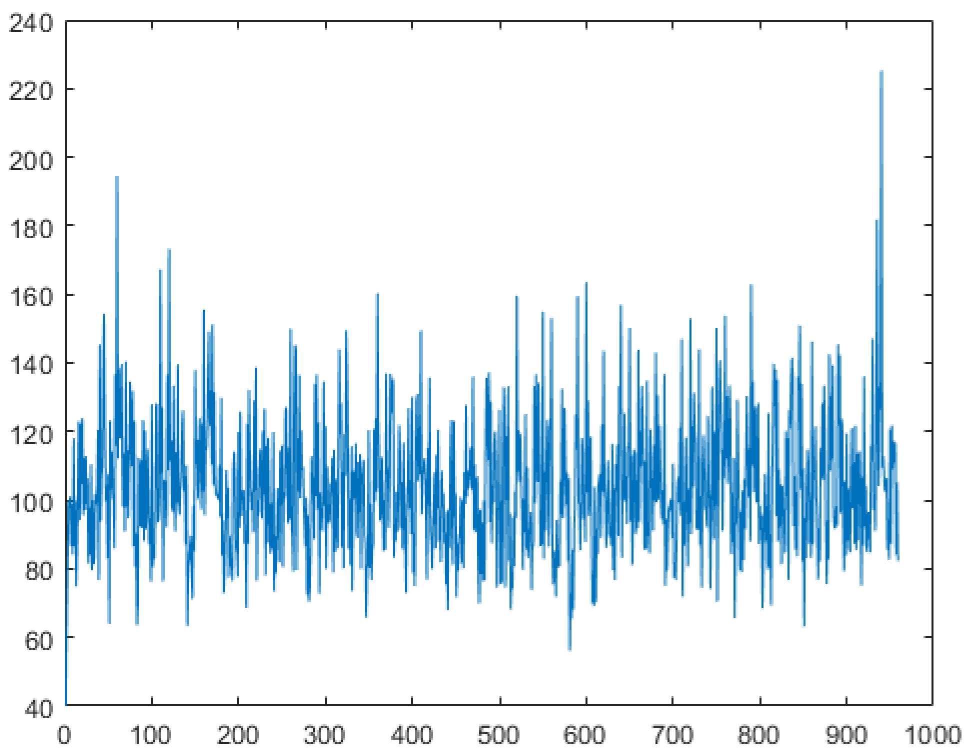


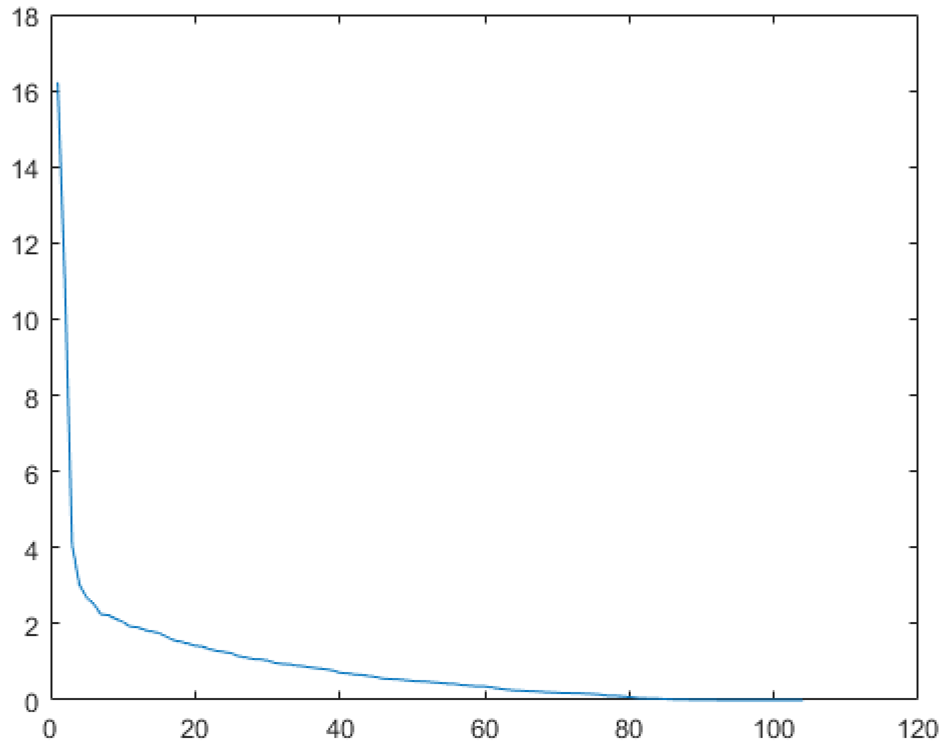
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8

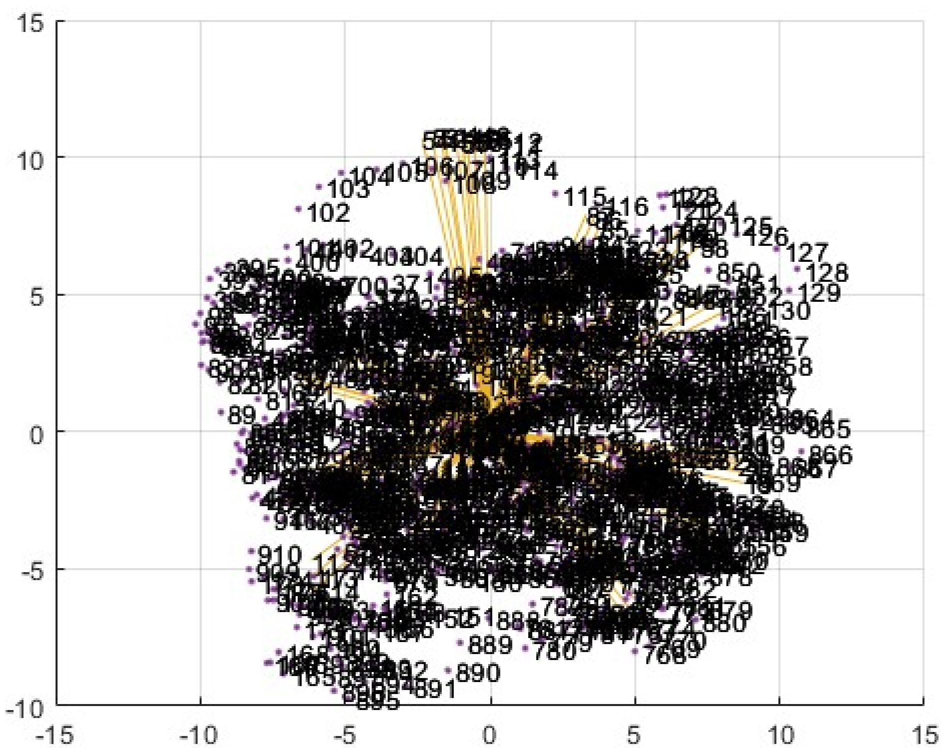

90
80
70
60
50
40
30
20
10
0 100 200 300 400 500 600 700 800 900 1000

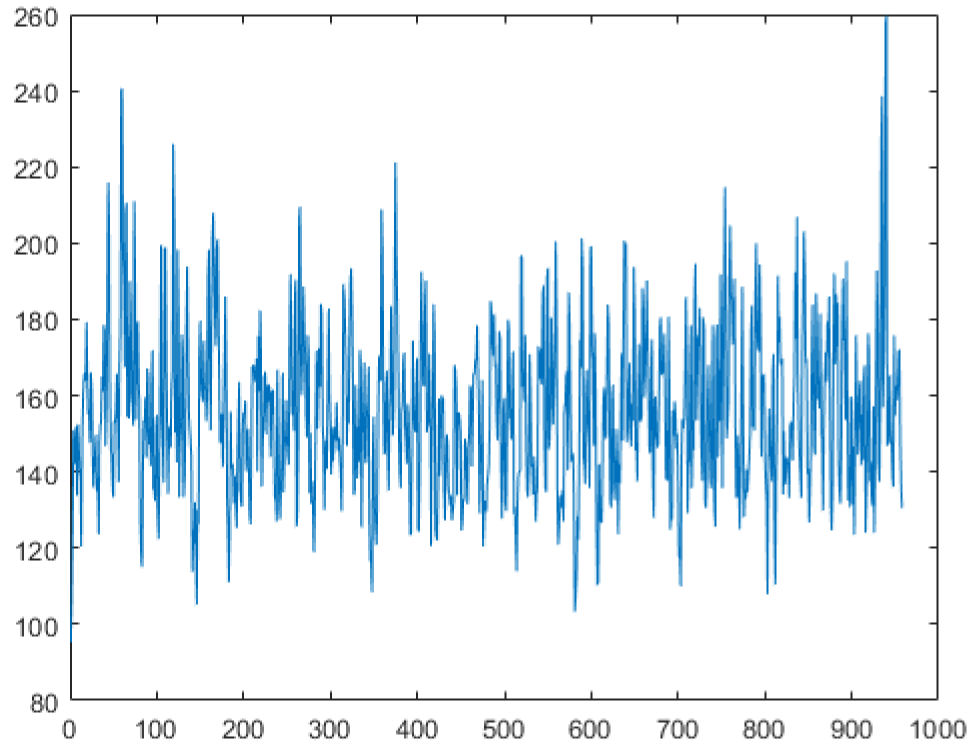

18
16
14
12
10
8
6
4
2
0
0 10 20 30 40 50 60


10
8
6
4
2
0
-2
-4
-6
-8
-10
-10 -8 -6 -4 -2 0 2 4 6 8 10

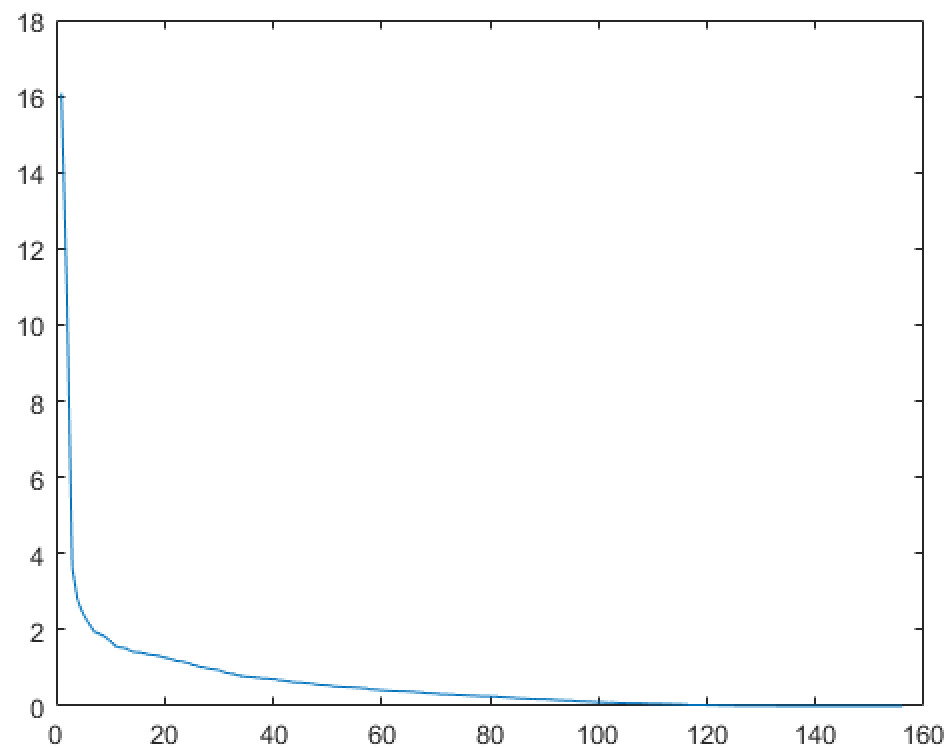

240
220
200
180
160
140
120
100
80
60
40
0 100 200 300 400 500 600 700 800 900 1000


0
2
4
6
8
10
12
14
16
18
0
20
40
60
80
100
120

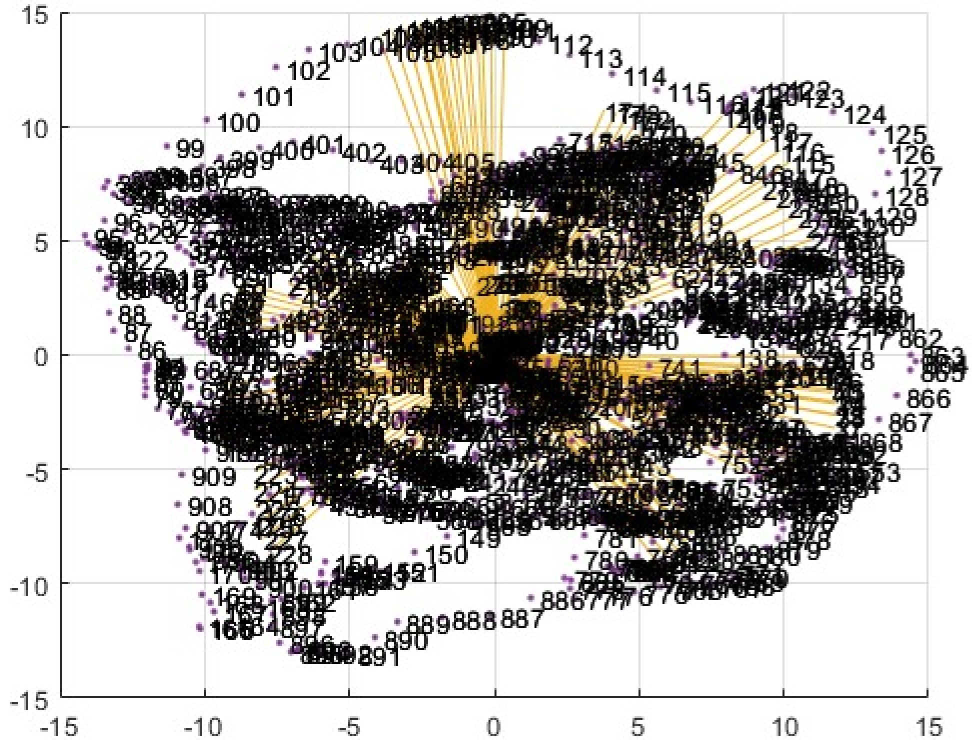

-10
-5
0
5
10
15
-15
-10
-5
0
5
10
15

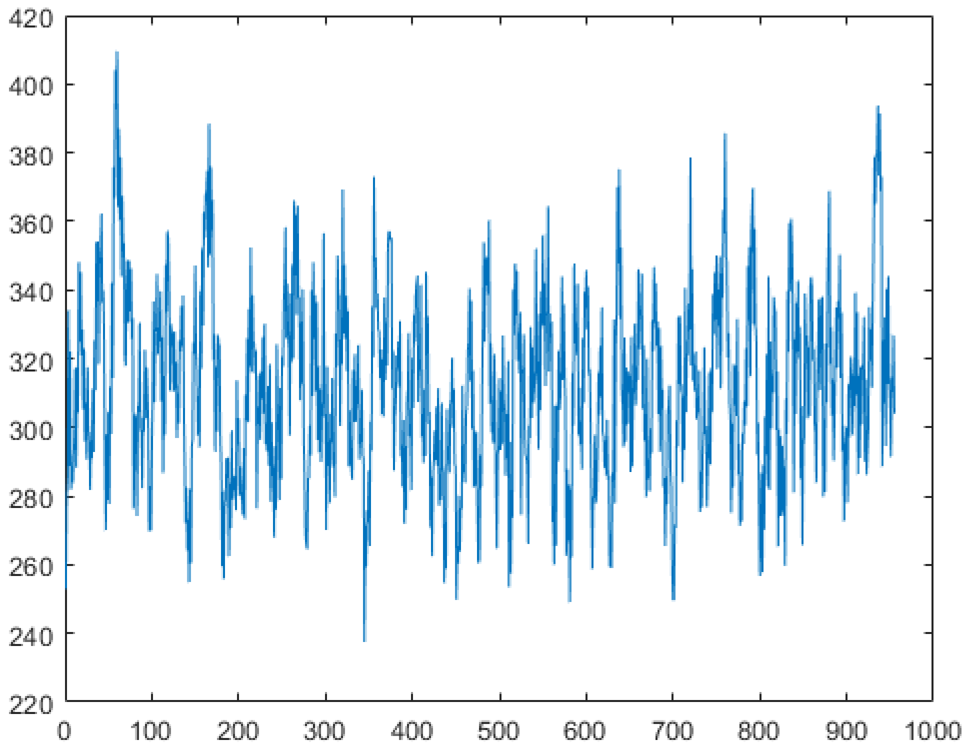


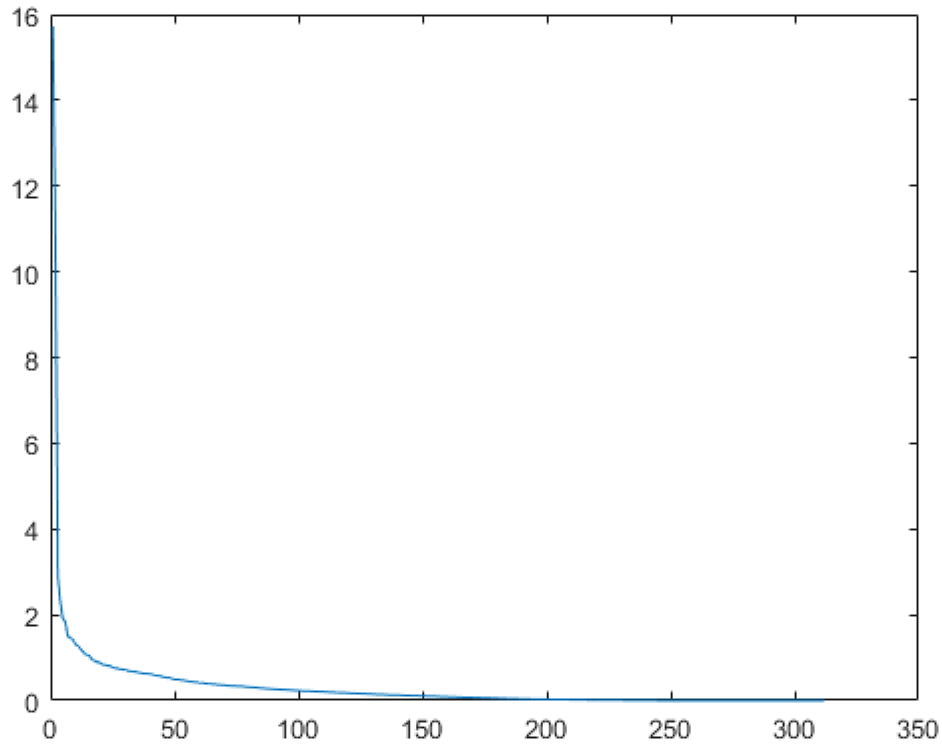
260
240
220
200
180
160
140
120
100
80
0
100
200
300
400
500
600
700
800
900
1000



18
16
14
12
10
8
6
4
2
0
0
20
40
60
80
100
120
140
160


15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
15


420
400
380
360
340
320
300
280
260
240
220
0
100
200
300
400
500
600
700
800
900
1000

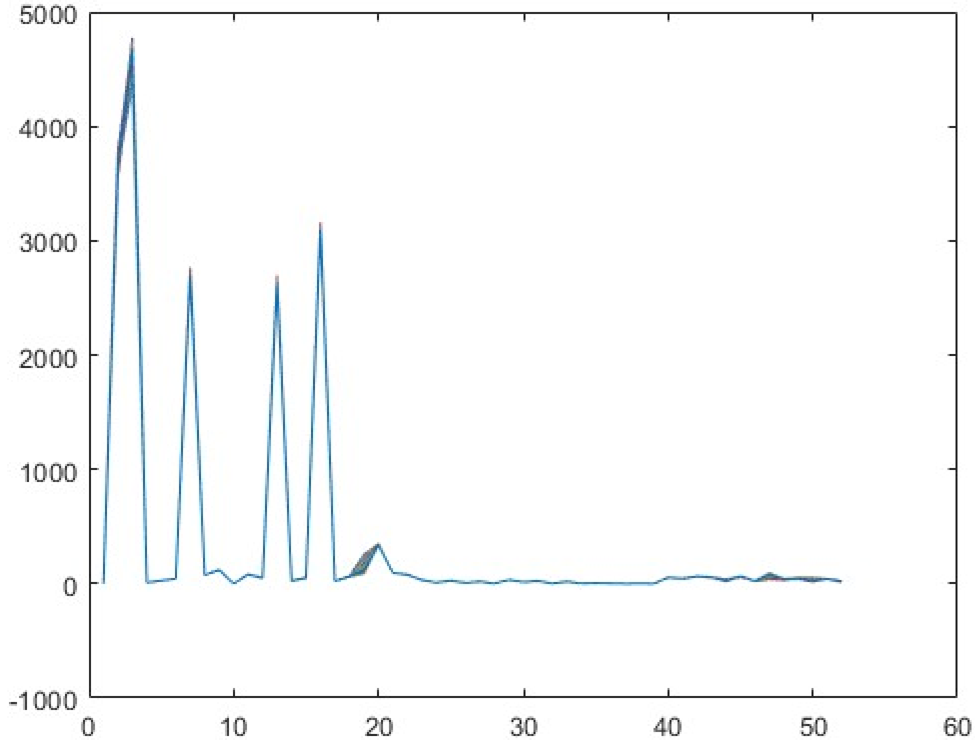


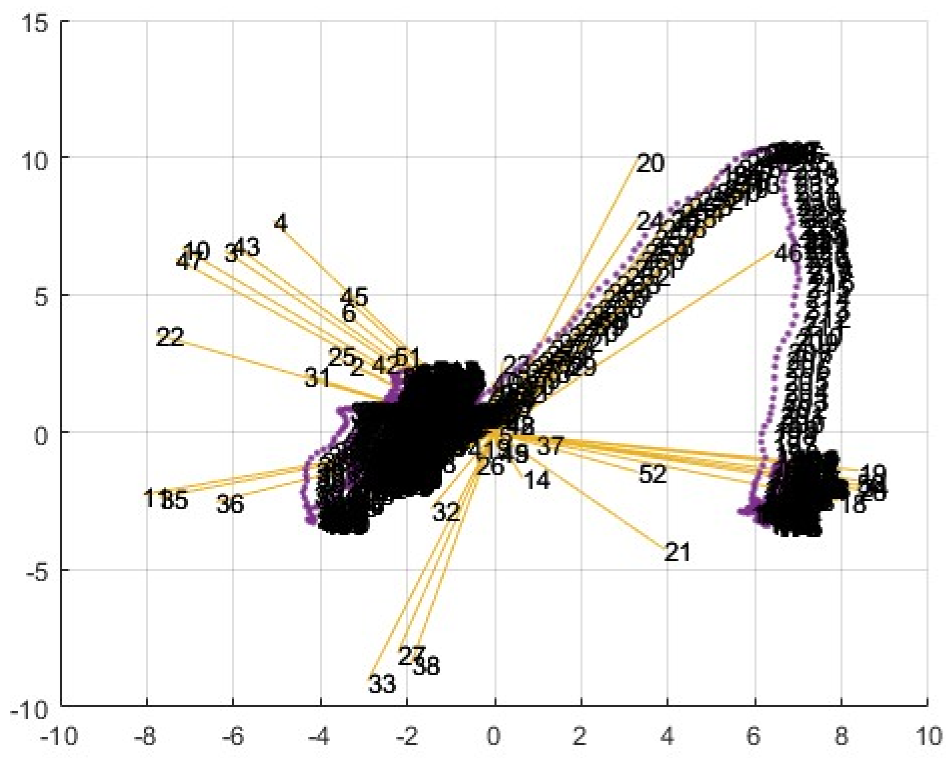
ST0532623: Case 3: (data set 2- d02_te)

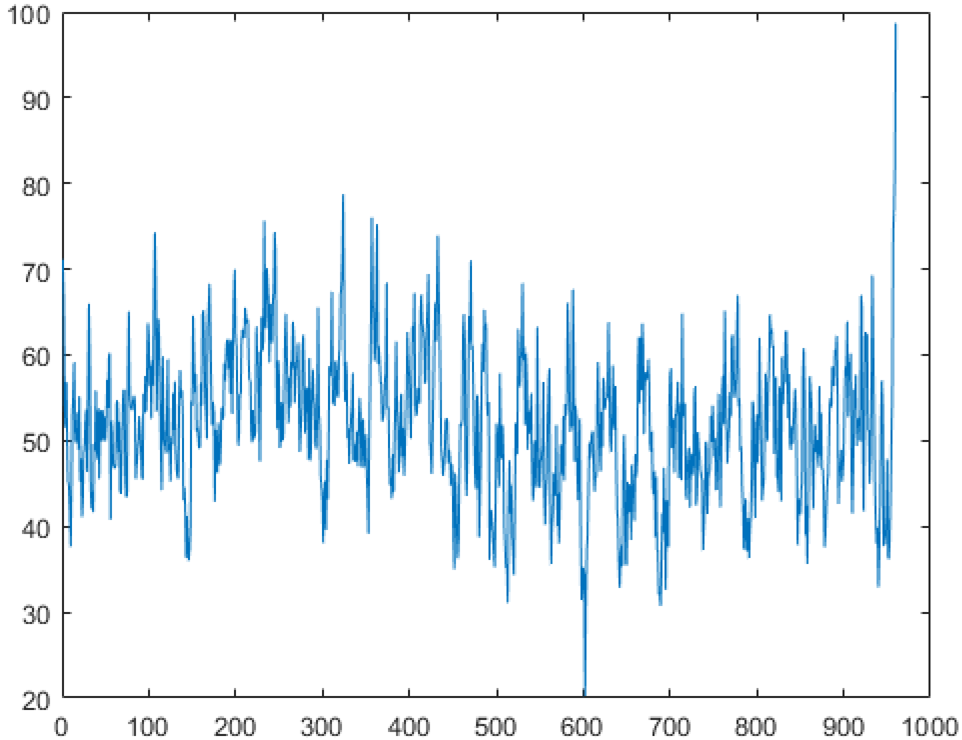
All the process are same as previous. So I am not commenting all the lines.

```
load ('d02_te.dat')
figure(33)
plot(1:52,d02_te)
% normalize the data
normalize_data_2=normalize(d02_te);
data_centered_original_2 = center(maverage((normalize_data_2),13,0,1));
[coeff2,score2,latent2,tsquared2,explained2]=pca((data_centered_original_2)); % done pca
figure (34); hold on; grid on;
plotbi(score2,coeff2,1,2,1:size(data_centered_original_2,1),[],1:size(score2,1),1:size(score2,2));
figure(35); plot(tsquared2)
figure(36);plot(explained2)
% coefficient of determination
% rsquared2=cumsum(explained2)

%DPCA
% Create the new matrix with lag 1
lag_mat2=time_lag_matrix(d02_te,1);
% normalize the new data
norm_lag_mat_2=normalize(lag_mat2);
```

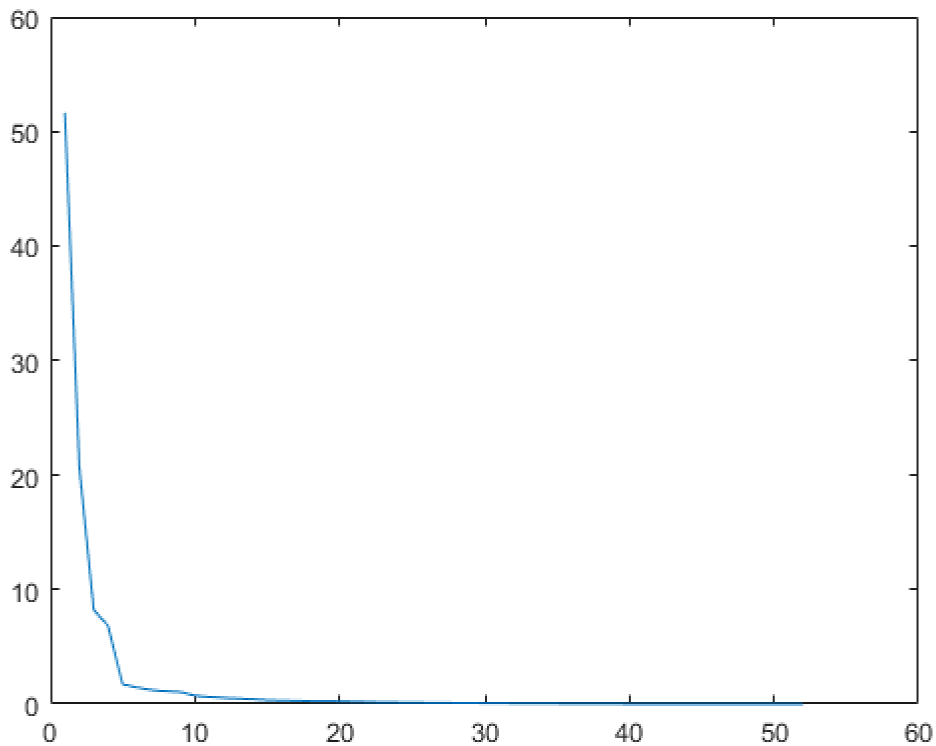
```
[coeff2_new,score2_new,latent2_new,tsquared2_new,explained2_new]=pca((norm_lag_mat_2));
figure (37); hold on; grid on;
plotbi(score2_new,coeff2_new,1,2,1:size(norm_lag_mat_2,1),[],1:size(score2_new,1),1:size(score2_new,2));
figure(38); plot(tsquared2_new)
figure(39);plot(explained2_new)
% rsquarednew2=cumsum(explained2_new)

% Create the new matrix with lag 2
lag_mat2_2=time_lag_matrix(d02_te,2);
% normalize the new data
norm_lag_mat_2_2=normalize(lag_mat2_2);
[coeff2_new_2,score2_new_2,latent2_new_2,tsquared2_new_2,explained2_new_2]=pca((norm_lag_mat_2_2));
figure (40); hold on; grid on;
plotbi(score2_new_2,coeff2_new_2,1,2,1:size(norm_lag_mat_2_2,1),
[],1:size(score2_new_2,1),1:size(score2_new_2,2));
figure(41); plot(tsquared2_new_2)
figure(42);plot(explained2_new_2)

% Create the new matrix with lag 3
lag_mat2_3=time_lag_matrix(d02_te,3);
% normalize the new data
norm_lag_mat_2_3=normalize(lag_mat2_3);
[coeff2_new_3,score2_new_3,latent2_new_3,tsquared2_new_3,explained2_new_3]=pca((norm_lag_mat_2_3));
figure (43); hold on; grid on;
plotbi(score2_new_3,coeff2_new_3,1,2,1:size(norm_lag_mat_2_3,1),
[],1:size(score2_new_3,1),1:size(score2_new_3,2));
figure(44); plot(tsquared2_new_3)
figure(45);plot(explained2_new_3)

% Create the new matrix with lag 6
lag_mat2_6=time_lag_matrix(d02_te,6);
% normalize the new data
norm_lag_mat_2_6=normalize(lag_mat2_6);
[coeff2_new_6,score2_new_6,latent2_new_6,tsquared2_new_6,explained2_new_6]=pca((norm_lag_mat_2_6));
figure (46); hold on; grid on;
plotbi(score2_new_6,coeff2_new_6,1,2,1:size(norm_lag_mat_2_6,1),
[],1:size(score2_new_6,1),1:size(score2_new_6,2));
figure(47); plot(tsquared2_new_6)
figure(48);plot(explained2_new_6)
```


5000
4000
3000
2000
1000
0
-1000
0
10
20
30
40
50
60

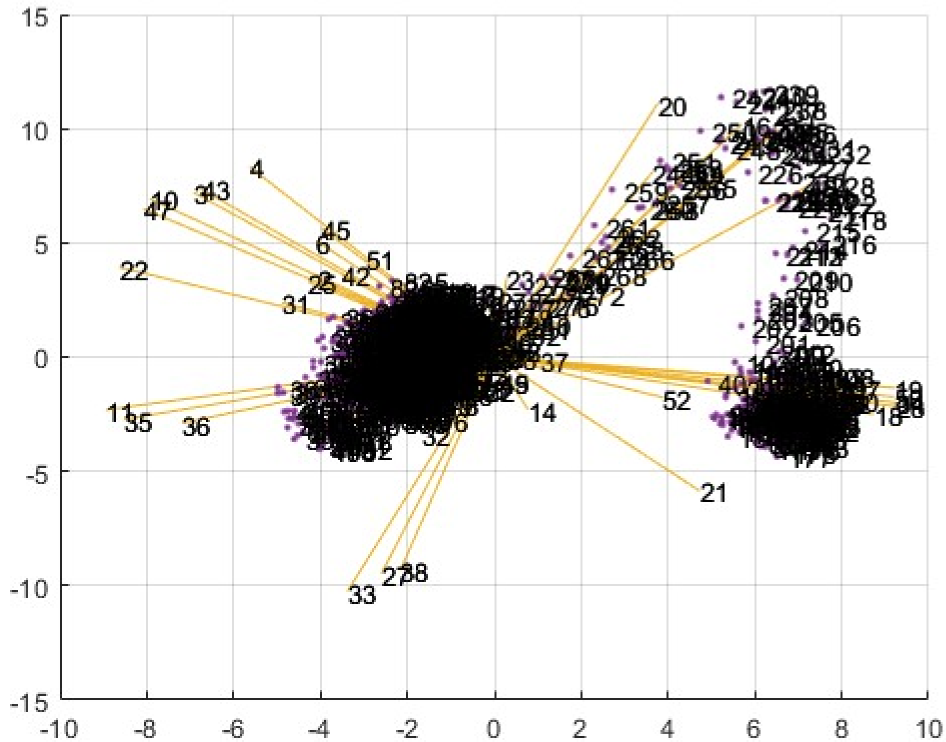

15
10
5
0
-5
-10
-10
-8
-6
-4
-2
0
2
4
6
8
10

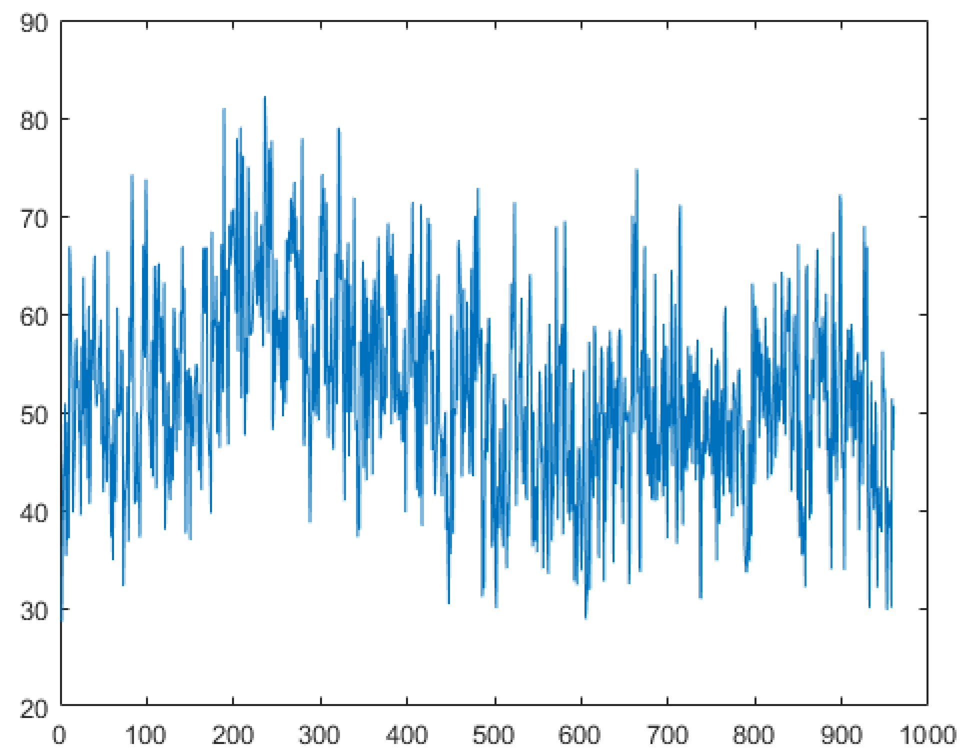


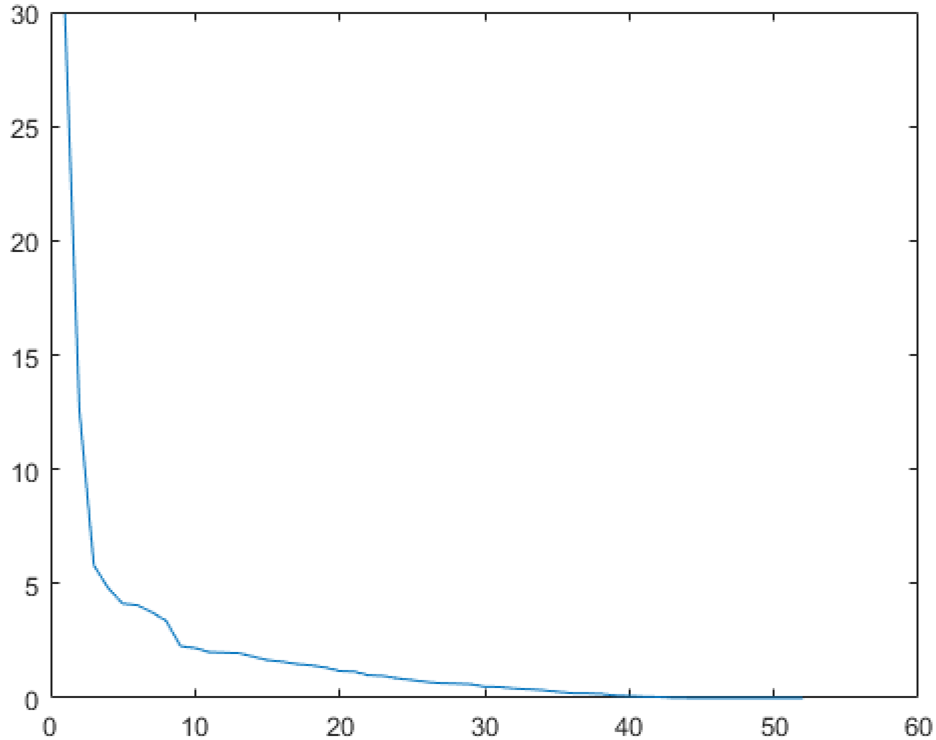
100
90
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
800
900
1000



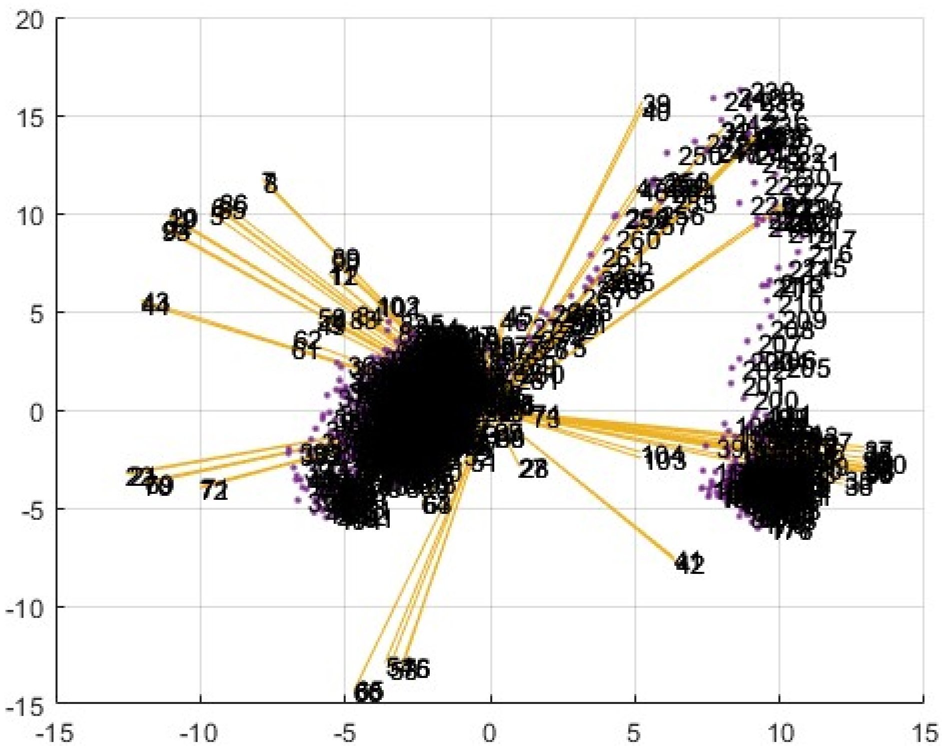
60
50
40
30
20
10
0
0
10
20
30
40
50
60


15
10
5
0
-5
-10
-15
-10
-8
-6
-4
-2
0
2
4
6
8
10

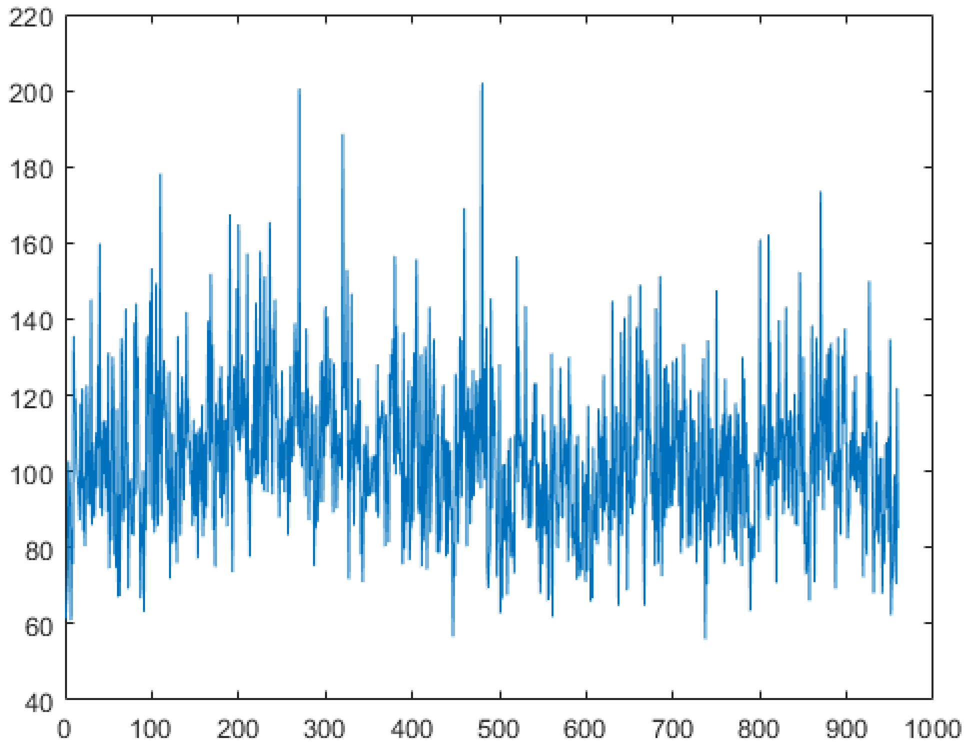

90
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
800
900
1000

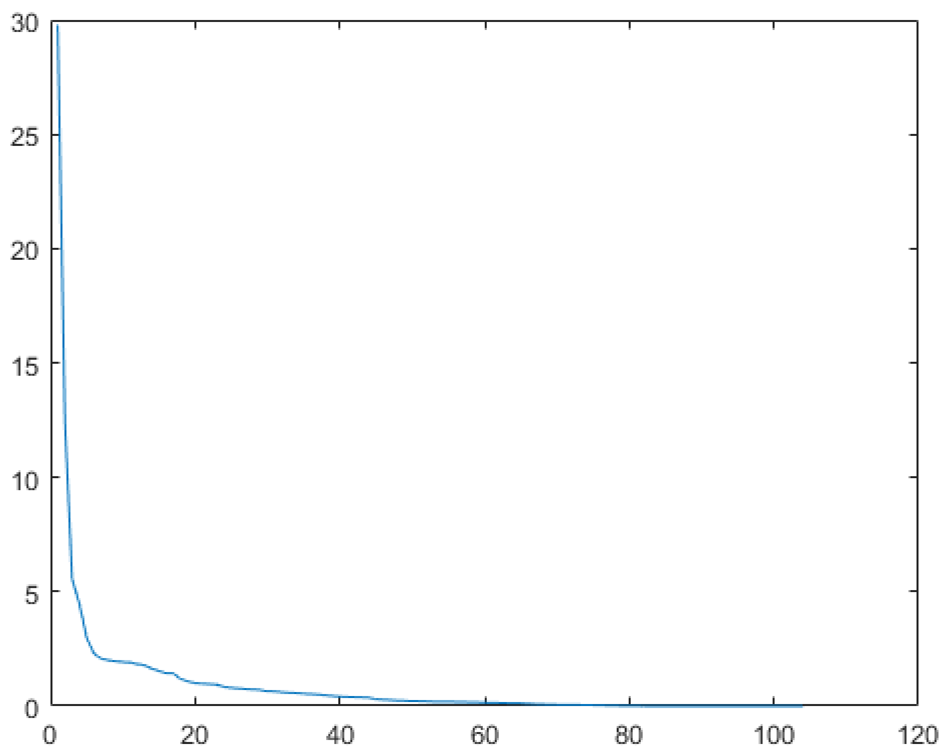


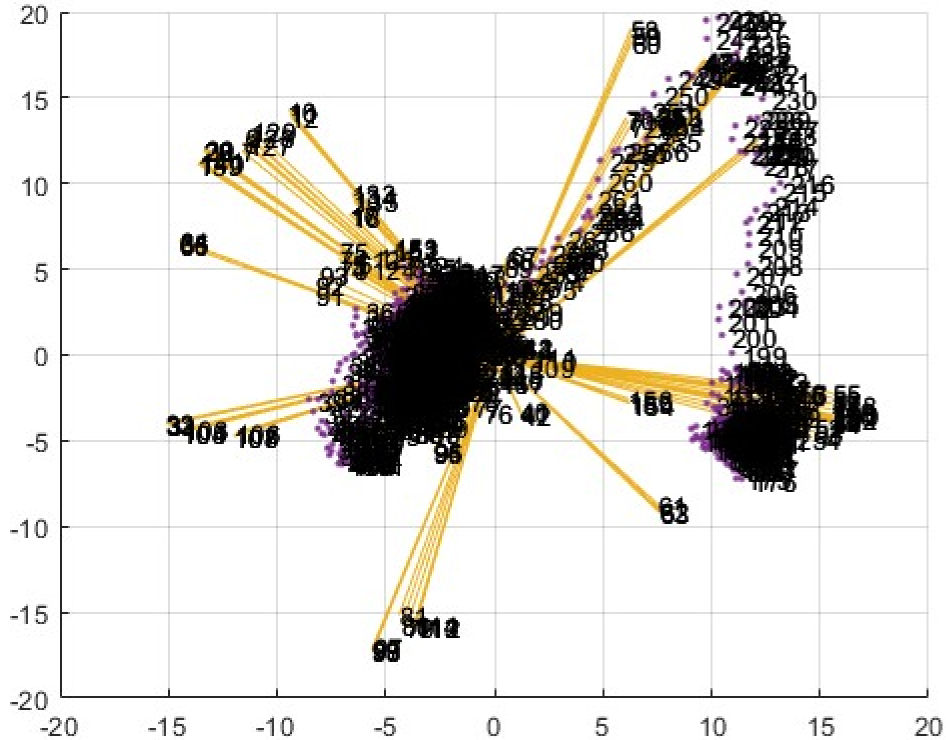
0
5
10
15
20
25
30
0
10
20
30
40
50
60

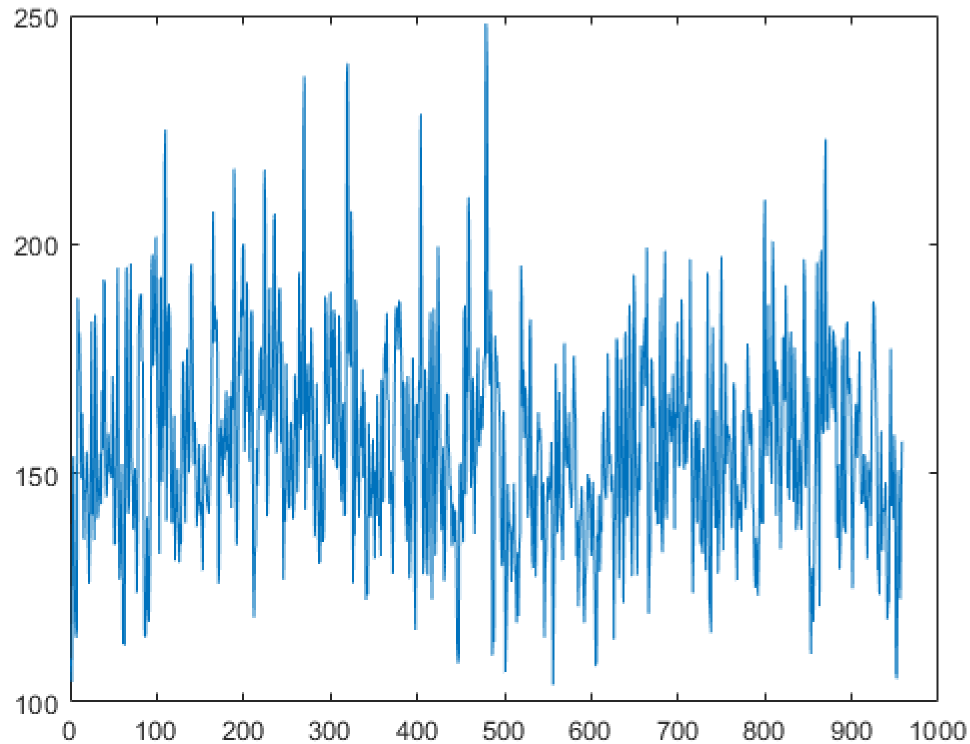


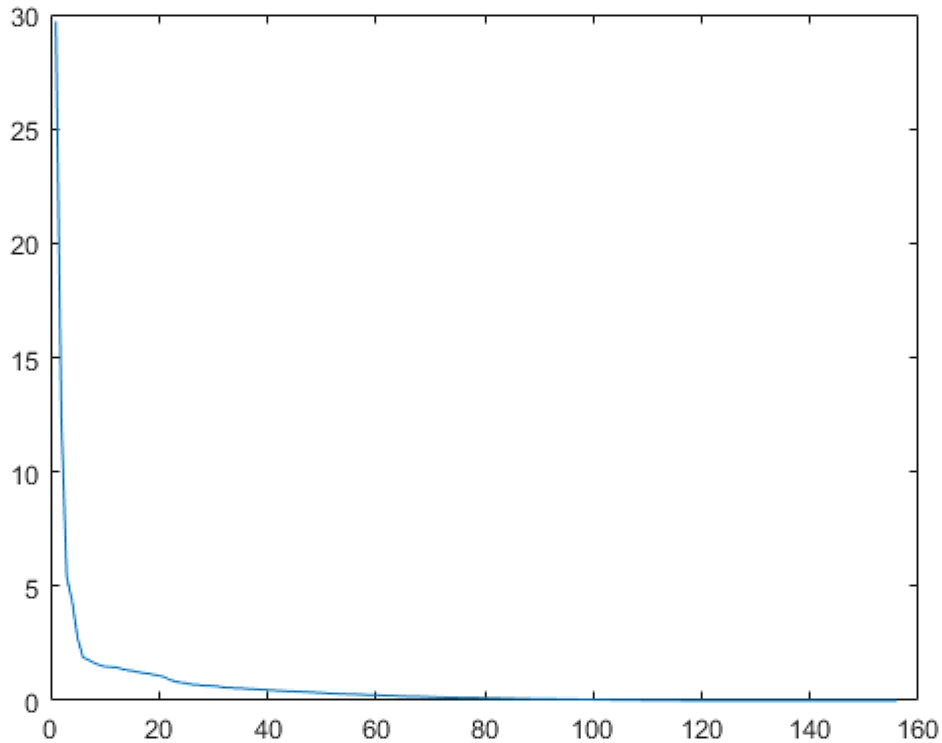
-15
-10
-5
0
5
10
15
20
-15
-10
-5
0
5
10
15


220
200
180
160
140
120
100
80
60
40
0
100
200
300
400
500
600
700
800
900
1000

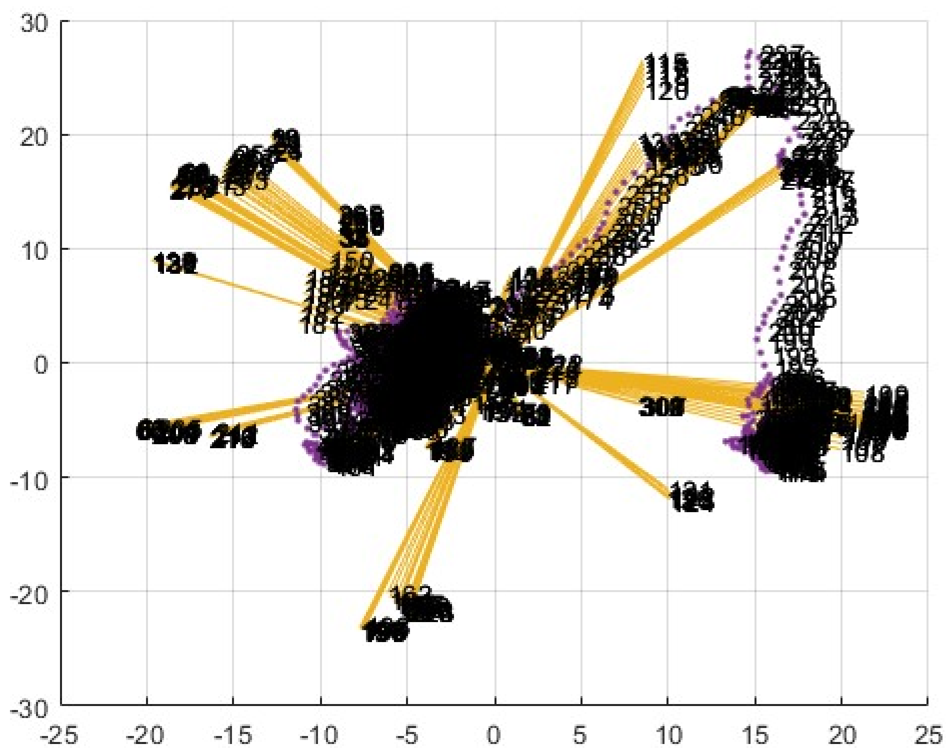

30
25
20
15
10
5
0
0
20
40
60
80
100
120


20
15
10
5
0
-5
-10
-15
-20
-20
-15
-10
-5
0
5
10
15
20

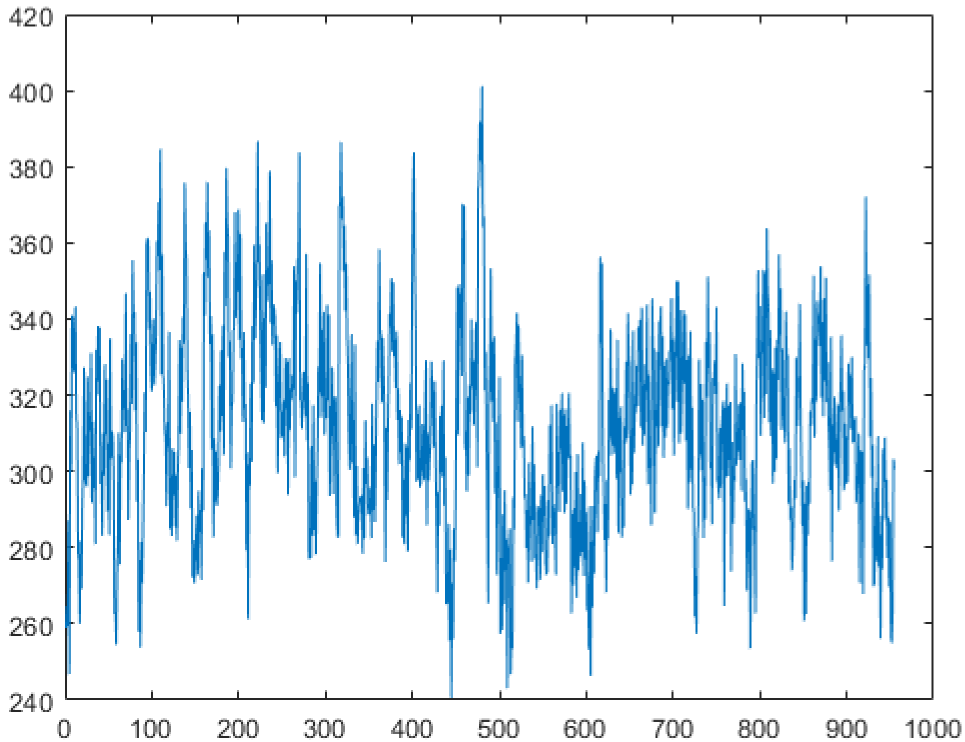

250
200
150
100
0
100
200
300
400
500
600
700
800
900
1000



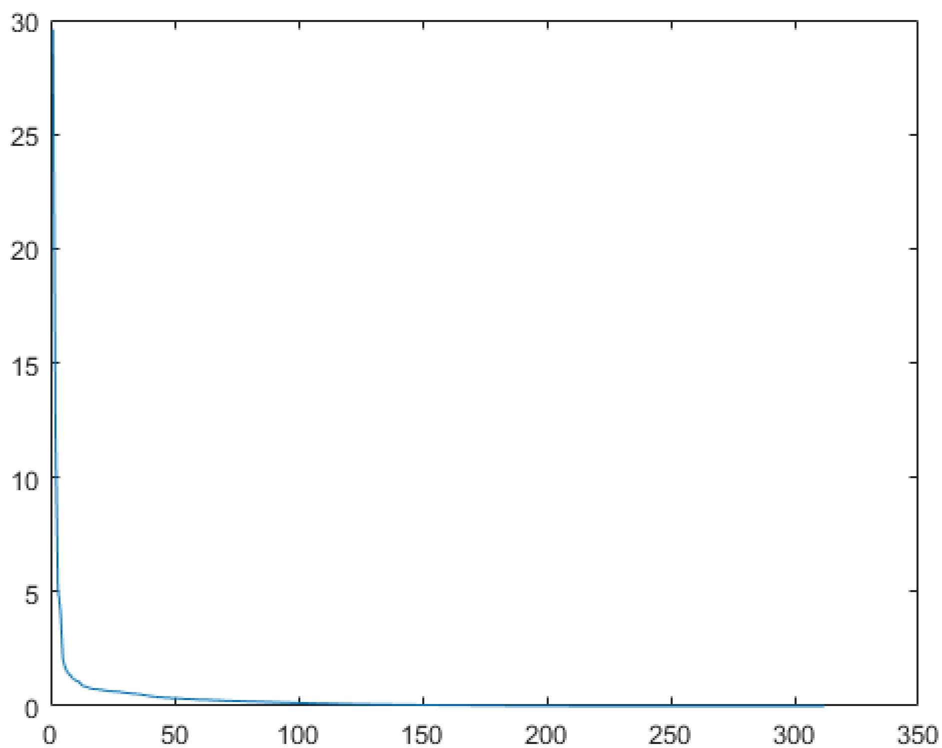
30
25
20
15
10
5
0
0
20
40
60
80
100
120
140
160



30
20
10
0
-10
-20
-30
-25
-20
-15
-10
-5
0
5
10
15
20
25



420
400
380
360
340
320
300
280
260
240
0
100
200
300
400
500
600
700
800
900
1000



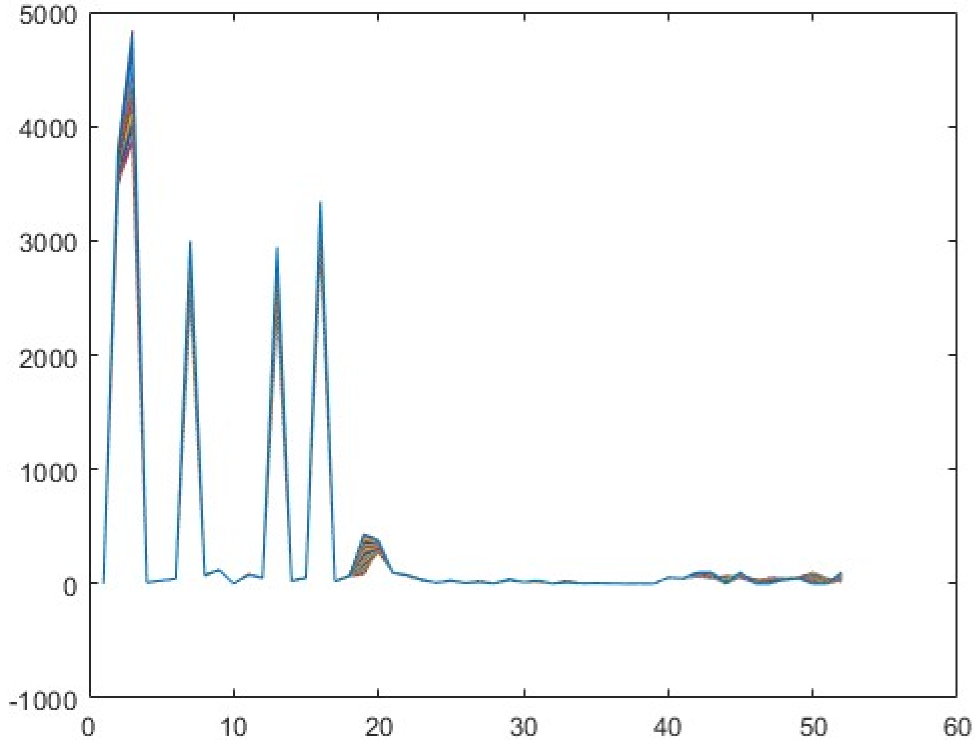
30
25
20
15
10
5
0
0
50
100
150
200
250
300
350

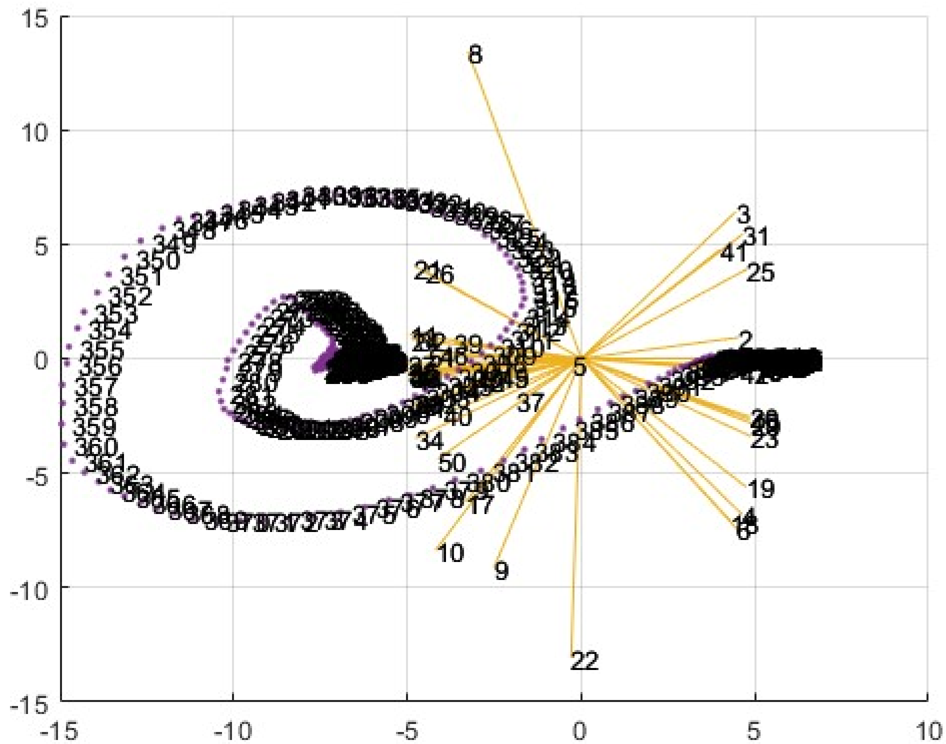
ST0532623: Case 4:(Data set 18-d18_te)

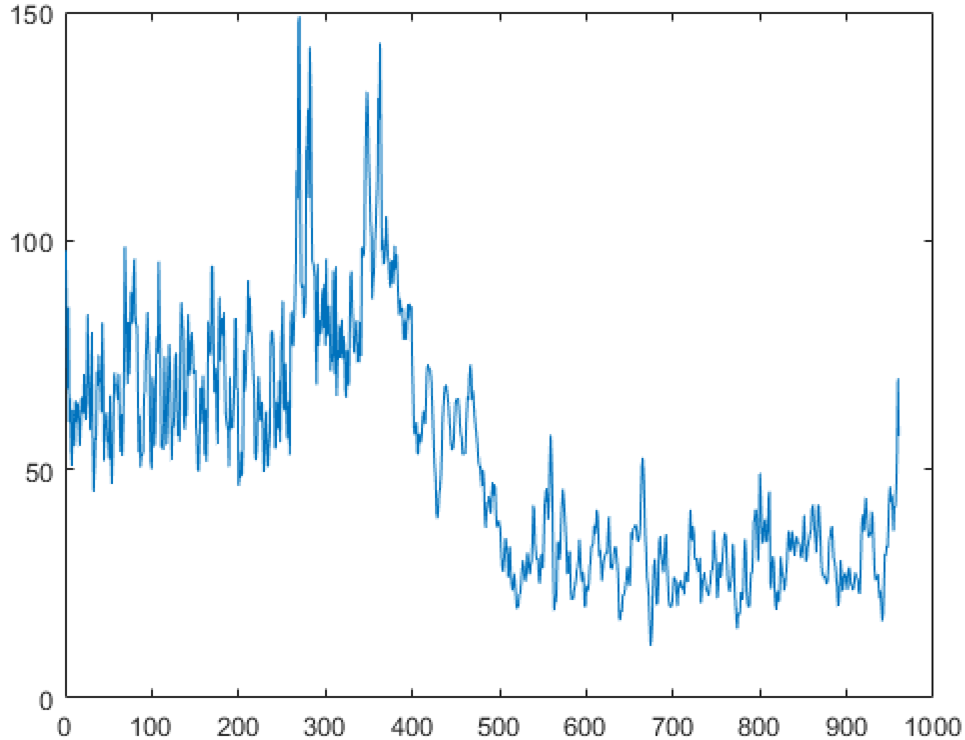
All the process are same as previous that's why i am not commenting all the lines as it is repetation.

```
load ('d18_te.dat')
figure(49)
plot(1:52,d18_te)
% normalize the data
normalize_data_18=normalize(d18_te);
data_centered_original_18 = center(maverage((normalize_data_18),13,0,1));
[coeff18,score18,latent18,tsquared18,explained18]=pca((data_centered_original_18));
figure (50); hold on; grid on;
plotbi(score18,coeff18,1,2,1:size(data_centered_original_18,1),[],1:size(score18,1),1:size(score18,2));
figure(51); plot(tsquared18)
figure(52);plot(explained18)
% coefficient of determination
% rsquared2=cumsum(explained2)
%DPCA
% Create the new matrix with lag 1
lag_mat18=time_lag_matrix(d18_te,1);
% normalize the new data
norm_lag_mat_18=normalize(lag_mat18);
[coeff18_new,score18_new,latent18_new,tsquared18_new,explained18_new]=pca((norm_lag_mat_18));
figure (53); hold on; grid on;
plotbi(score18_new,coeff18_new,1,2,1:size(norm_lag_mat_18,1),[],1:size(score18_new,1),1:size(score18_new,2));
figure(54); plot(tsquared18_new)
figure(55);plot(explained18_new)
% rsquarednew2=cumsum(explained2_new)

% Create the new matrix with lag 2
lag_mat18_2=time_lag_matrix(d18_te,2);
% normalize the new data
norm_lag_mat_18_2=normalize(lag_mat18_2);
[coeff18_new_2,score18_new_2,latent18_new_2,tsquared18_new_2,explained18_new_2]=pca((norm_lag_mat_18_2)
);
figure (56); hold on; grid on;
plotbi(score18_new_2,coeff18_new_2,1,2,1:size(norm_lag_mat_18_2,1),
[],1:size(score18_new_2,1),1:size(score18_new_2,2));
figure(57); plot(tsquared18_new_2)
figure(58);plot(explained18_new_2)

% Create the new matrix with lag 3
lag_mat18_3=time_lag_matrix(d18_te,3);
% normalize the new data
norm_lag_mat_18_3=normalize(lag_mat18_3);
[coeff18_new_3,score18_new_3,latent18_new_3,tsquared18_new_3,explained18_new_3]=pca((norm_lag_mat_18_3)
```

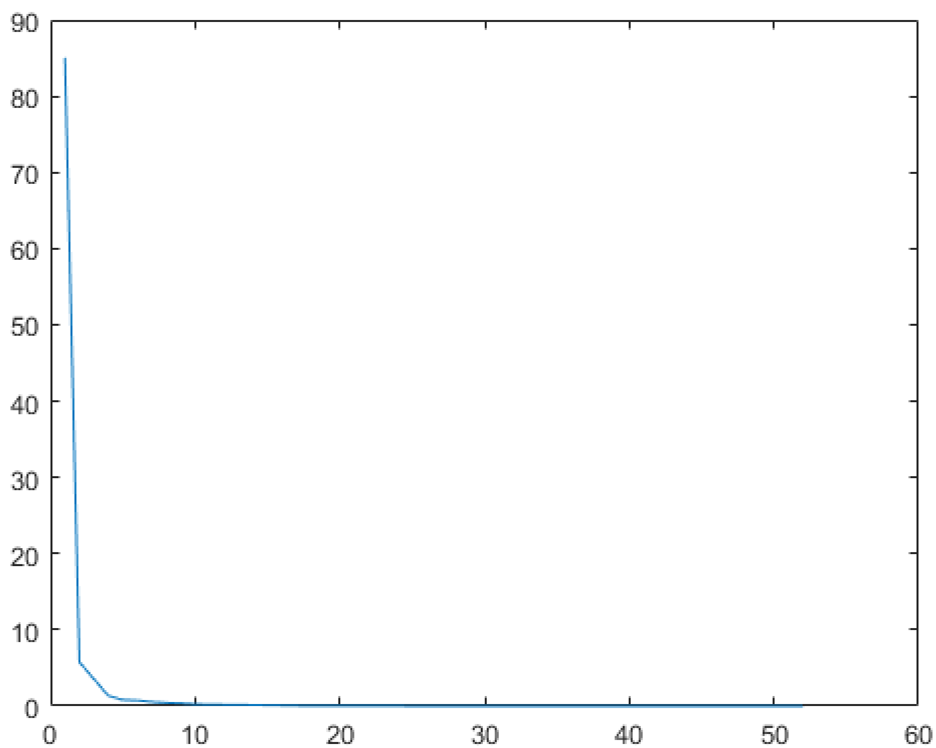
```
);
figure (59); hold on; grid on;
plotbi(score18_new_3,coeff18_new_3,1,2,1:size(norm_lag_mat_18_3,1),
[],1:size(score18_new_3,1),1:size(score18_new_3,2));
figure(60); plot(tsquared18_new_3)
figure(61);plot(explained18_new_3)

% Create the new matrix with lag 6
lag_mat18_6=time_lag_matrix(d18_te,6);
% normalize the new data
norm_lag_mat_18_6=normalize(lag_mat18_6);
[coeff18_new_6,score18_new_6,latent18_new_6,tsquared18_new_6,explained18_new_6]=pca((norm_lag_mat_18_6)
);
figure (62); hold on; grid on;
plotbi(score18_new_6,coeff18_new_6,1,2,1:size(norm_lag_mat_18_6,1),
[],1:size(score18_new_6,1),1:size(score18_new_6,2));
figure(63); plot(tsquared18_new_6)
figure(64);plot(explained18_new_6)

% Reference for 'center' and 'maverage' code: Teaching assistant has
% provided sample solution in Exercise 3 task 5. It was uploded in the
% moodle.
```


5000
4000
3000
2000
1000
0
-1000
0
10
20
30
40
50
60

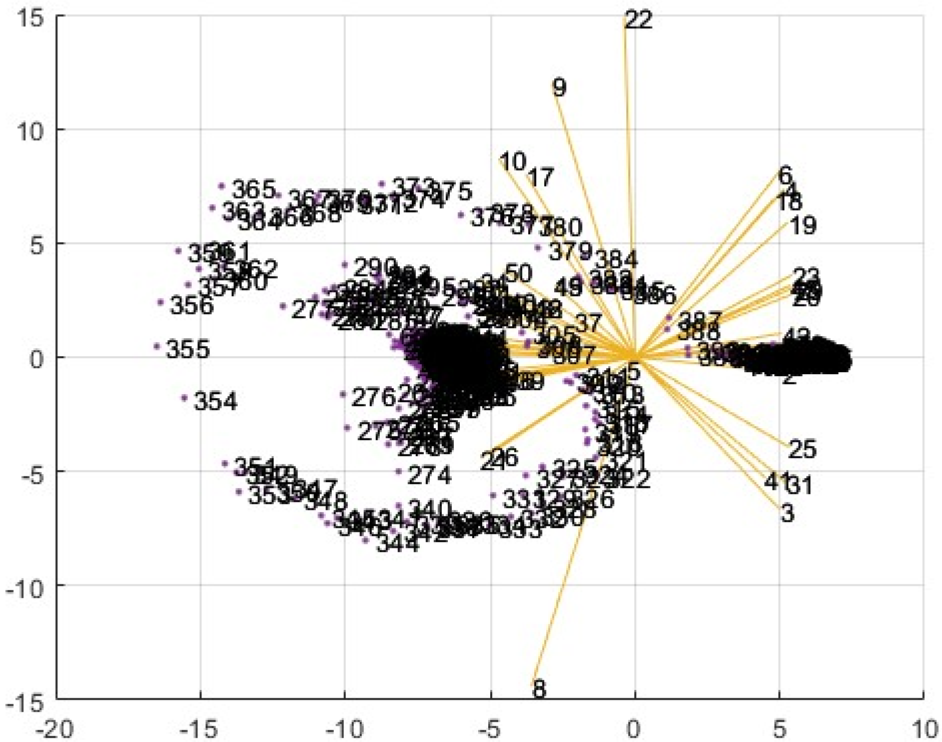

15
10
5
0
-5
-10
-15
-15
-10
-5
0
5
10
8
3
31
41
25
2
5
37
34
40
50
10
9
22
19
23

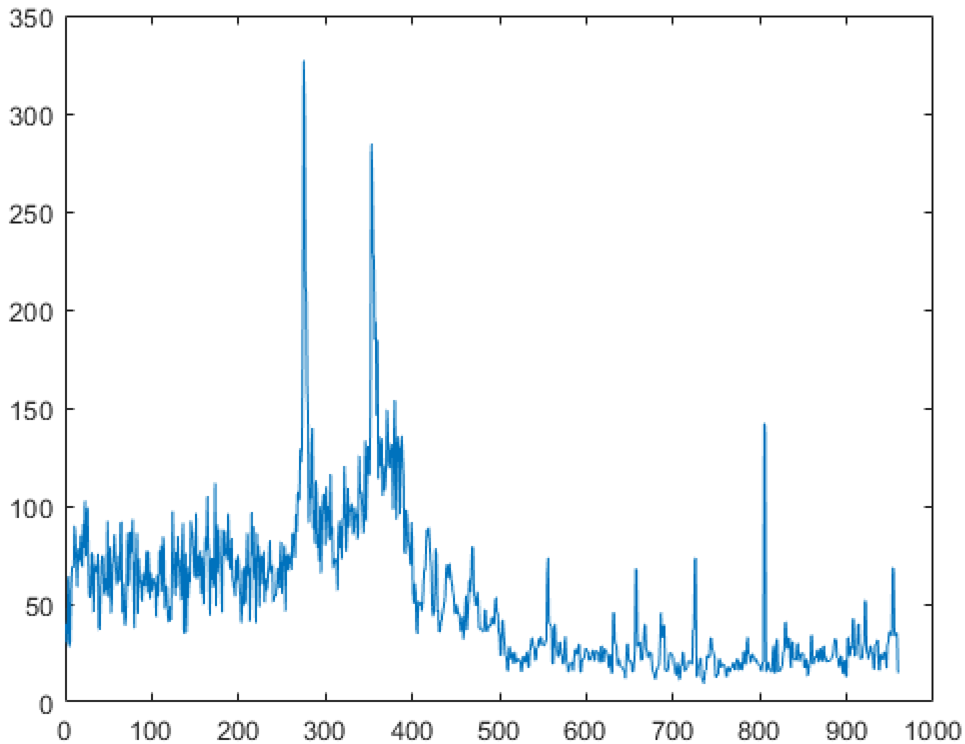


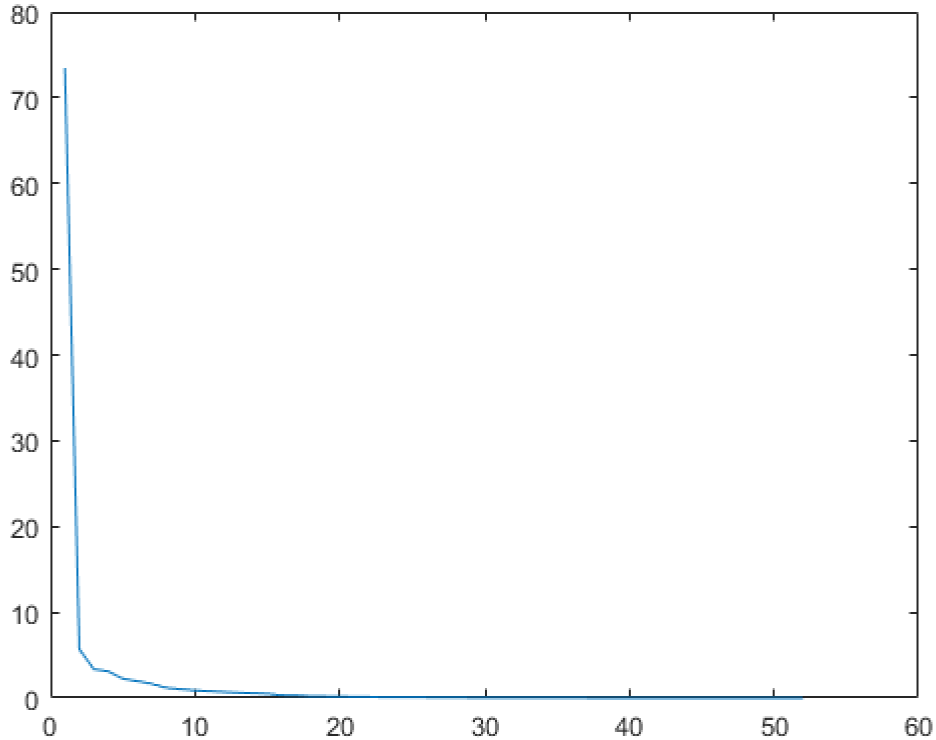
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1000

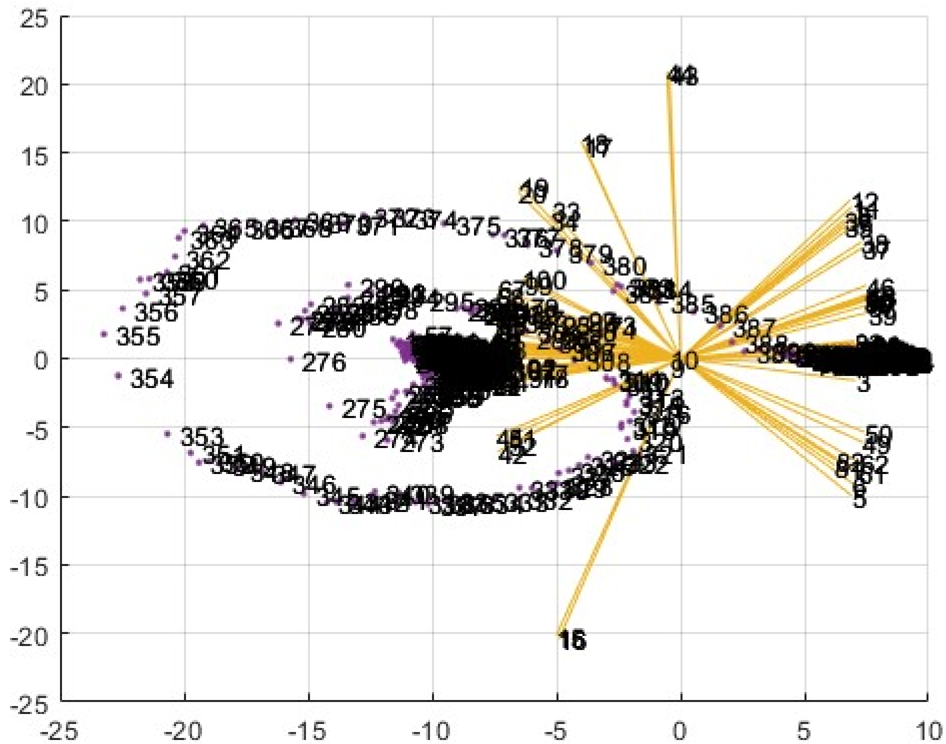


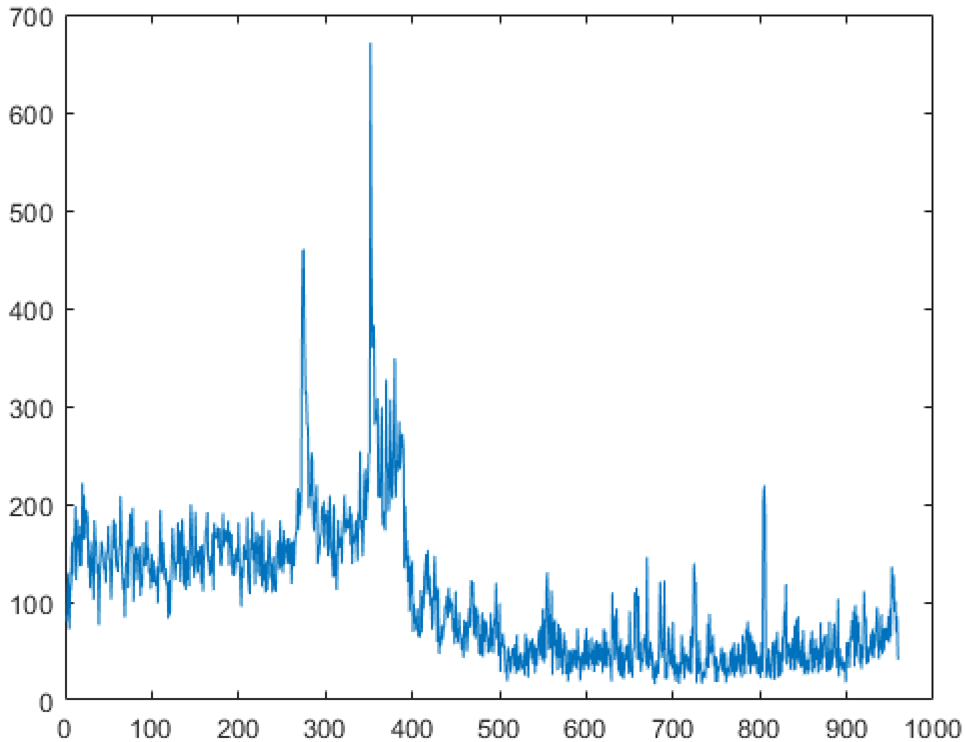
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60

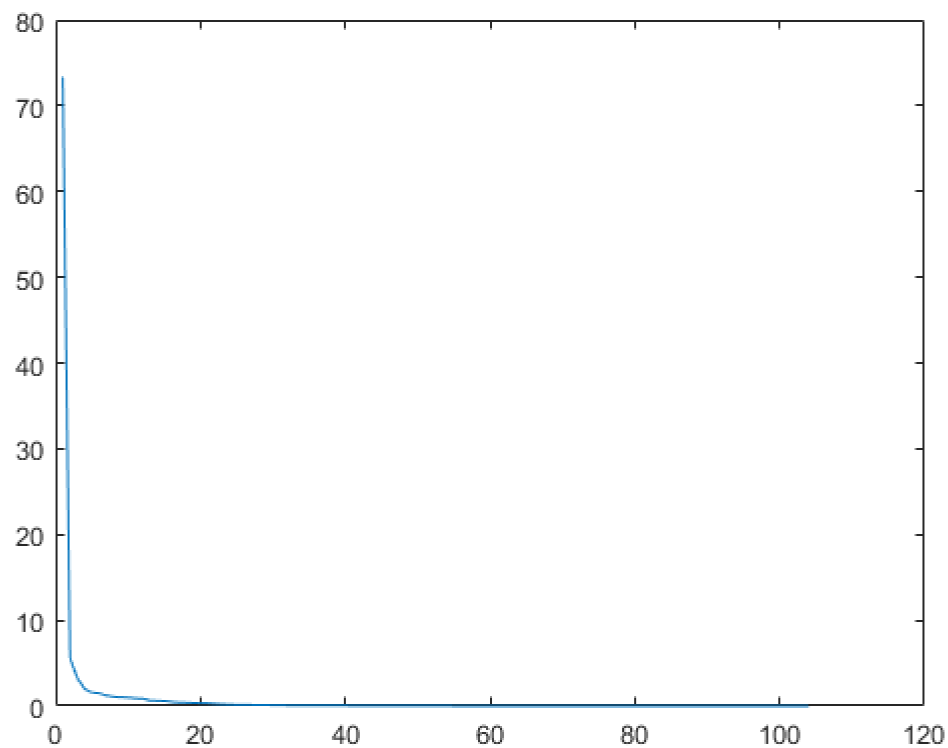

15
10
5
0
-5
-10
-15
-20
-15
-10
-5
0
5
10

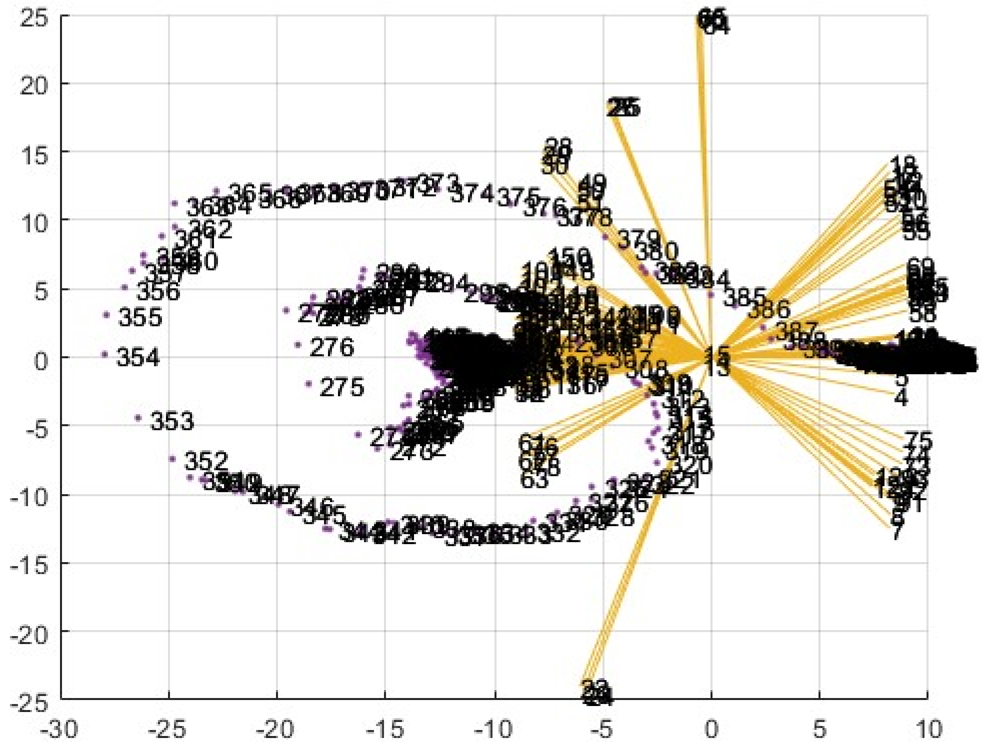

350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
1000

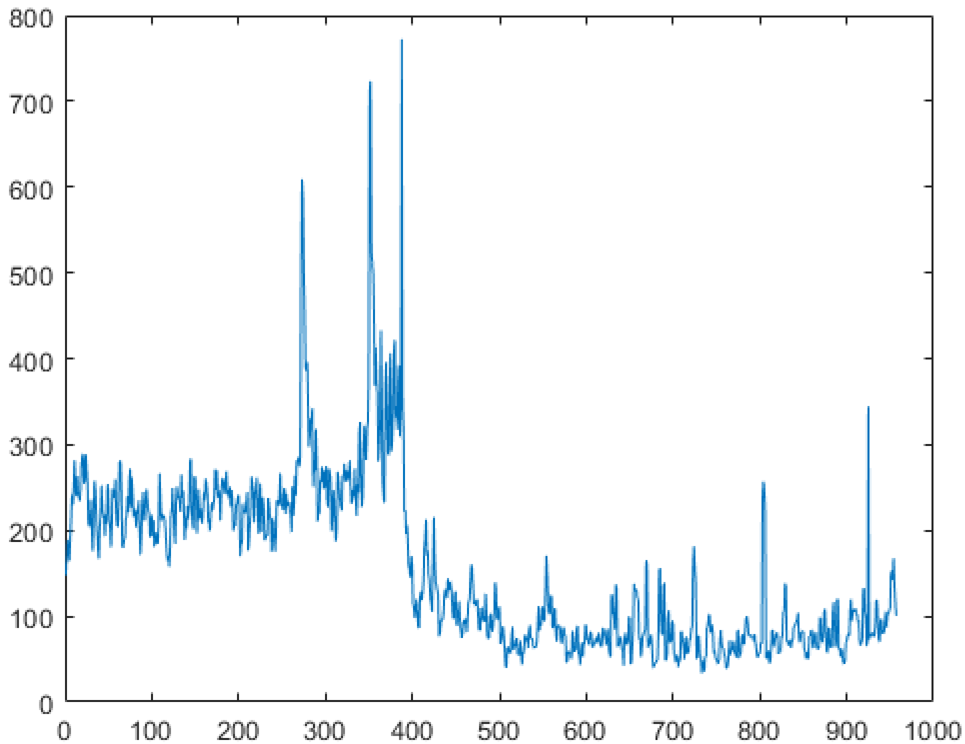

80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60

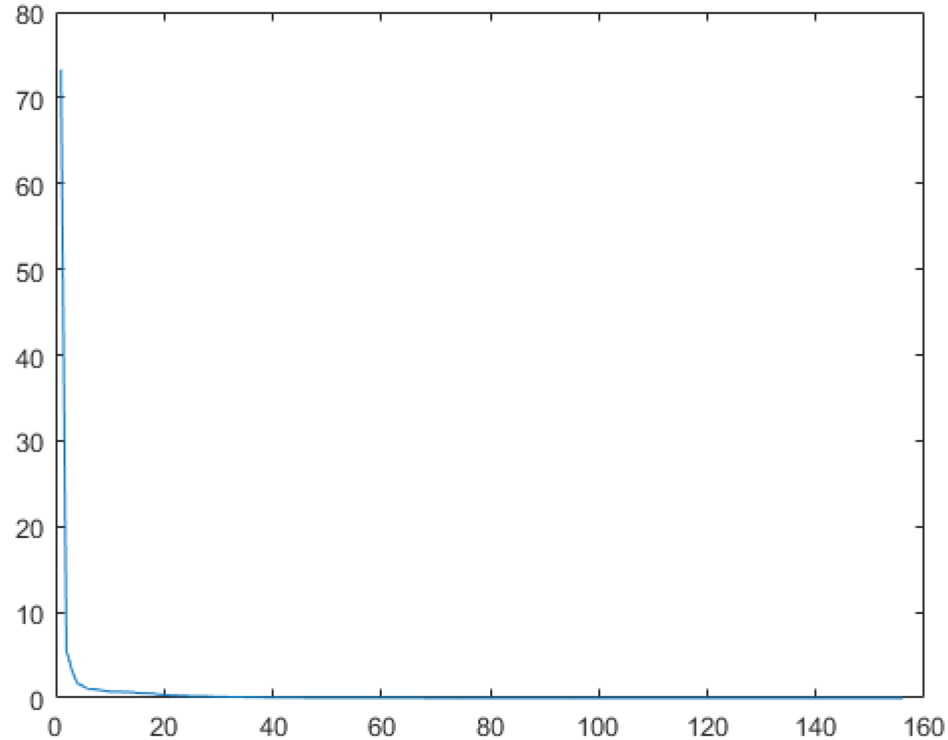

25
20
15
10
5
0
-5
-10
-15
-20
-25
-25
-20
-15
-10
-5
0
5
10


700
600
500
400
300
200
100
0
0 100 200 300 400 500 600 700 800 900 1000

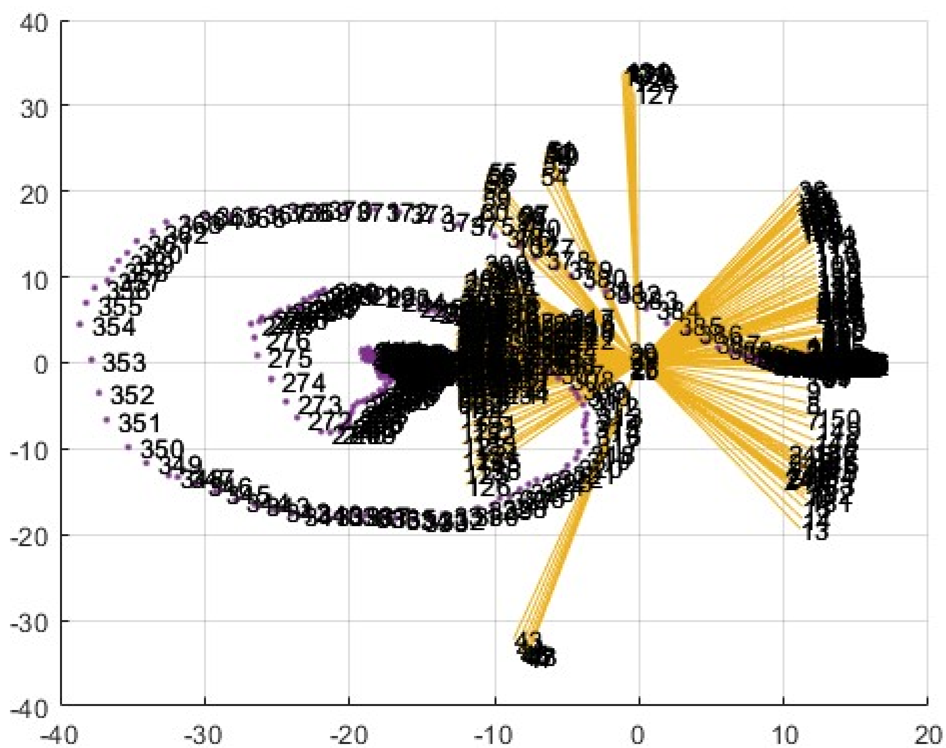

80
70
60
50
40
30
20
10
0
0 20 40 60 80 100 120


25
20
15
10
5
0
-5
-10
-15
-20
-25
-30
-25
-20
-15
-10
-5
0
5
10

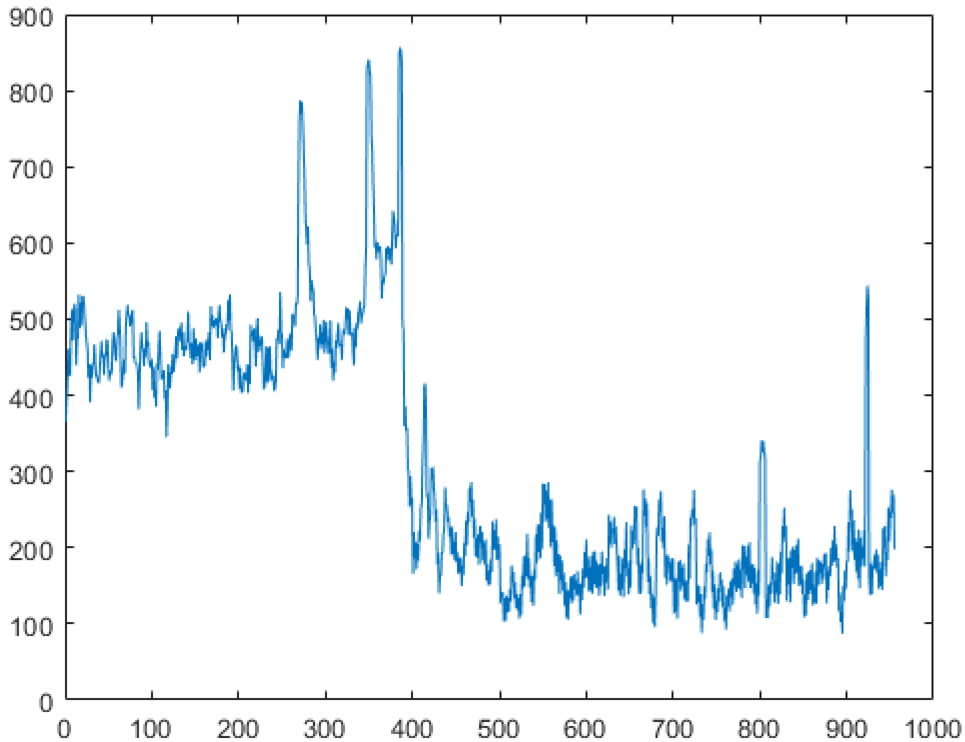

800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000

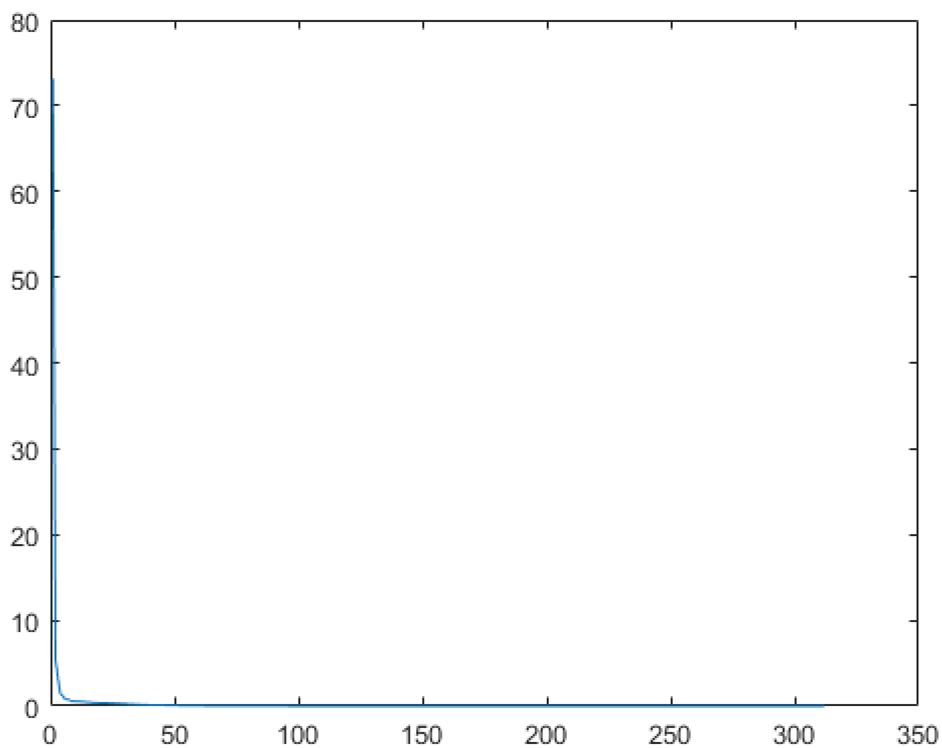


80
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120
140
160



40
30
20
10
0
-10
-20
-30
-40
-40
-30
-20
-10
0
10
20


900
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000


80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
300
350

Published with MATLAB® R2018b