BM20A6100 Advanced Data Analysis and Machine Learning

Project name: « Fault detection in an industrial process»

Autors
Student: Dmitrii Shumilin
Student number: 0589870
E-mail: ShumilinDmAl@gmail.com
Dmitrii.Shumilin@student.lut.fi

23.10.2020

1. Introduction

Failure prediction based on time series is an important step in the control of the production process. These time series in industry are most often getting information from sensors, counters, control signals, communication signals and flags. Tracking these variables helps understand if the manufacturing process is within its normal range of operations.

The production process can deviate from normal operation for a number of possible reasons:

- Aging and degradation of equipment
- Errors in the operation of equipment personnel
- Malicious attacks on production processes in order to disable the process or lead to irreversible consequences for people and the environment

In this project, deviations from the norm are considered for the last two reasons. They can be characterized by a sufficiently fast (abrupt effect on parameters) having either a short-term effect, when the equipment, under the influence of its own control system, reaches the nominal mode, or long-term, when the arisen malfunctions continue to exist in time.

2. Dataset

In our case, we will investigate several cases from the Tennessee dataset. The Tennessee Eastman dataset was created by the Eastman Chemical Company to simulate a manufacturing process. They created a digital twin of the chemical industry object, which produces 52 time series with the ability to simulate emergency situations. In the original dataset, based on work (Russel et al. 2000), there are 22 different cases of system behavior, one of which corresponds to the nominal operating mode.

3. Choosing fault detection method

The choice of a method for fault detecting in the data is based on the purpose of the study and the available data. Data mining methods can be broken down into supervised learning and unsupervised learning. Supervised learning is based on the fact that we know in advance the answers to the detection task, that is, whether there is an anomaly in the data or not, the start time of the anomalous process and the time of its end. Unsupervised detection methods can be used when there is no such information available. In our case, there is no markup at the beginning or end of the anomalous process, so we will refer this task to the unsupervised tasks.

Performing unsupervised fault detection can be performed based on hand based rules, statistical methods and machine learning algorithms.

Hand based rules imply that there will be a person who has experience in the work of the process under study and knows possible approaches to solving this problem. This person establishes simple rules (inequalities) based on a causal relationship, the violation of which is considered as abnormal behavior of the process. The disadvantages of this approach are the following points:

- The need for a specialist
- Errors in hand based rules
- The time of integration the rules in industry
- Can only detect the behavior that the specialist has foreseen

Statistical methods are based on the use of classical statistics algorithms and statistical tests for getting significant statistical difference between the two values. This approach can work in real time and with a limited amount of data. The main disadvantages can be the fact that for the application of such algorithms it is often necessary to make some assumptions about the independence of variables or data points, also about the type of data distribution, which, obviously, is often not fulfilled.

Machine learning algorithms perform very well in this area. One of the most promising types of anomaly detection are detectors based on neural networks, for example, an RNN autoencoder is presented in work (Filonov et al. 2017). This approach is most accurate for detecting anomalies, according to the results of comparison of the authors of the article with statistical methods. Another obvious advantage of RNN networks is that it is possible to carry out anomaly detection in real time. The main disadvantage of this approach is that training neural networks requires a huge amount of data that is difficult and expensive to obtain.

In our case, we have a limited dataset of 960 measurements of 52 variables in normal operation. To solve the problem of detecting anomalies, we will use a statistical method.

4. Application for analysis of time series data

4.1 Class FaultDetector

The detector is based on the proposed detection methods in articles (Russel et al. 2000), (Wenfu Ku et al. 1995), namely PCA and DPCA decompositions and the calculation of T^2 and Q statistics. The work (Russel et al. 2000), implies the DPCA method - the application of PCA decomposition to the modified matrix of feature objects.

Preparing data for applying the methods takes several stages. The first is calculating the mean and standard deviation from the normal behavior of the variables. The second is to standardize a normal dataset. Third - applying the parameters obtained in the first step to standardize the values on a dataset with anomalous behavior

To select a new data dimension for PCA decomposition, we use the method of parallel analysis proposed in (Wenfu Ku et al. 1995). Let's generate a random dataset taken from a normal distribution of the same size as the original data. Apply SVD decomposition to noisy and data with anomaly behavior. We sort the obtained singular values in descending order and take the modulus of pairwise differences. The required future dimension in this case will be the index of the array with the smallest modulus of the pairwise difference. This process is automatically executed by the internal private function _get_a_number in the file “fault_detector.py”

Further, the calculation of the Hotelling or T^2 statistics is based on the formula from (Russel et al. 2000), after dividing the dataset by $\sqrt{n-1}$, where n is number of data points. Calculation formula for T^2 statistics (1).

$$T^2 = x^T P \Sigma_a^{-2} P^T x \quad (1)$$

Where: x – Vector of observations,

P – Loading vector related with a largest values,

Σ – Matrix which contain non negative a largest singular values.

Such calculation perform function “hotelling_statistic” in file “fault_detector.py”.

To calculate the Q statistics, we use the formula (2)

$$Q = \Sigma(X - XP_0P_0^T)^2 \tag{2}$$

Where: X – Dataset with anomaly behavior,

P_0 – Principal components from normal behavior dataset with corresponding a largest singular values.

Such calculation perform function “q_statistic” in file “fault_detector.py”

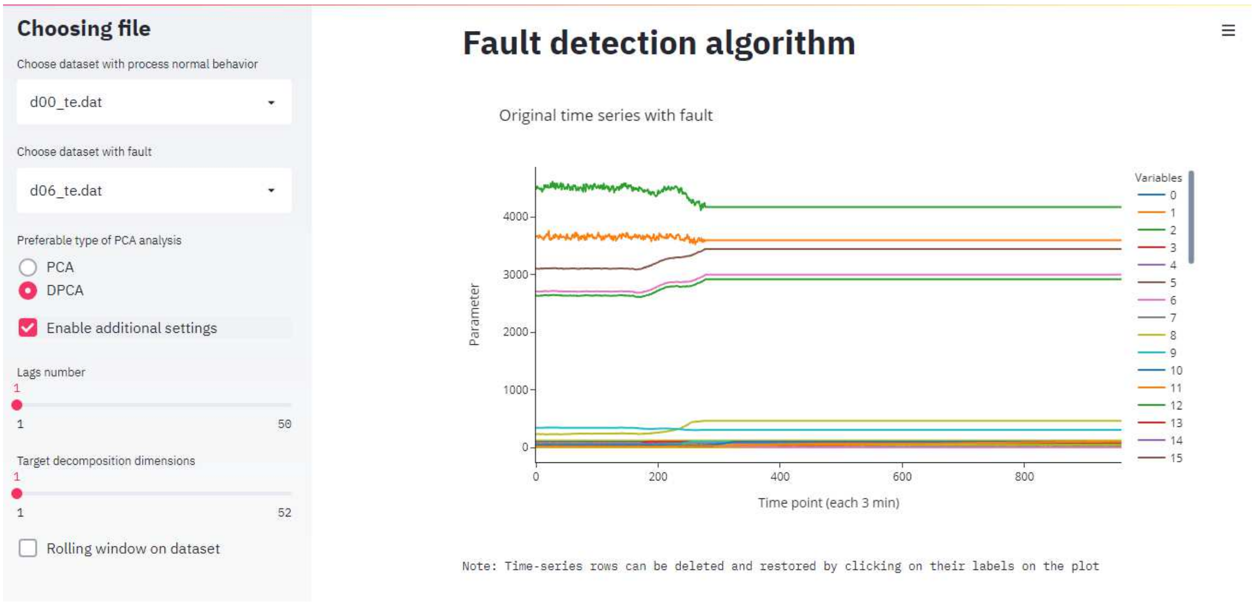
To perform DPCA and the corresponding augmentation of the dataset, it is necessary to determine the lag parameter L. In the current implementation, the L lag is determined automatically based on the algorithm proposed in (Wenfu Ku et al. 1995). The L selection algorithm is contained in the “_lag_num” function, the augmentation algorithm in the augmentation function in the “fault_detector.py” file.

Despite the fact that the automatic selection of the parameters a and L is implemented, these values can be entered manually when specifying a class for experimental purposes. Also, to reduce the effect of noise, it is possible to apply an averaging window on the original data by setting the corresponding flag to True and setting the window size.

4.2 Application

The application is entirely based on the streamlit python library, which runs it in a web browser. The library provides convenient widgets for interactive user interaction and displaying graphs from popular python libraries.

The appearance of the application is shown in Figure 1.



Choosing file

Choose dataset with process normal behavior

d00_te.dat

Choose dataset with fault

d06_te.dat

Preferable type of PCA analysis

○ PCA

● DPCA

☑ Enable additional settings

Lags number

1

1 50

Target decomposition dimensions

1

1 52

□ Rolling window on dataset

Fault detection algorithm

Original time series with fault

```
Note: Time-series rows can be deleted and restored by clicking on their labels on the plot
```

Figure 1. Overview of the appearance of the application interface

The application interface allows to freely select files for analysis. The user can select both existing data files and analyze their own. To do this, you need to upload files in the .mat format to the data folder located in the project root.

Time series analysis can be carried out in two different methods, PCA and DPCA. Despite the fact that the choice of the size of the new compressed feature space and the number of lags is made automatically, the user has the opportunity to set these parameters on his own by setting the flag next to the description “Enable addition settings”. Also, in the additional settings that appear, can be selected the previously described function of applying a moving average and set the window size.

Three types of interactive graphs are available to the user for analysis. Figure 2 depicts a basic time series that contains abnormal behavior. Time series rows from the plot can be hidden by clicking on the variable on the plot legend, and it is also possible to zoom in / out from the plot and select the necessary areas for research.

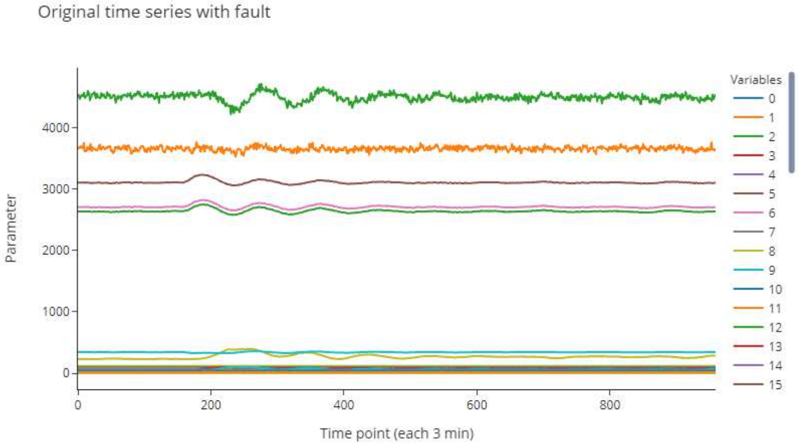


Figure 2. Original time series data with fault

Since most automatic methods for searching for anomalies do not allow distinguishing between variables in which anomalous behavior occurs, it is offered the opportunity to independently estimate the deviation from the norm the mean and standard deviation of the normalized dataset. This can serve as an indirect indication of in which variables the deviations from the normal behavior are greatest. This is the second available type of plot, shown in Figure 3. The plot is also interactive, when you hover the mouse over the column of interest in the histogram, you can get the number or name of a variable in the original dataset.

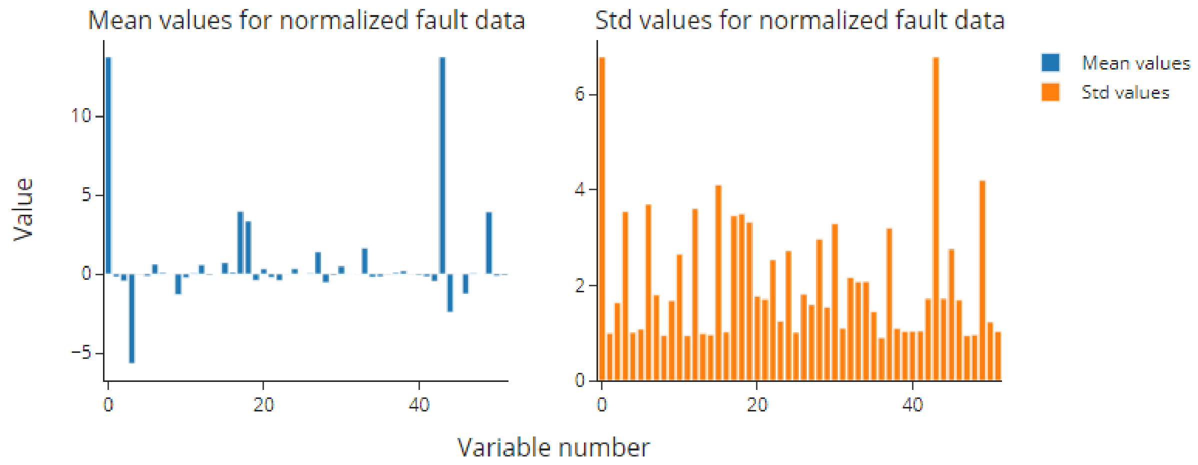
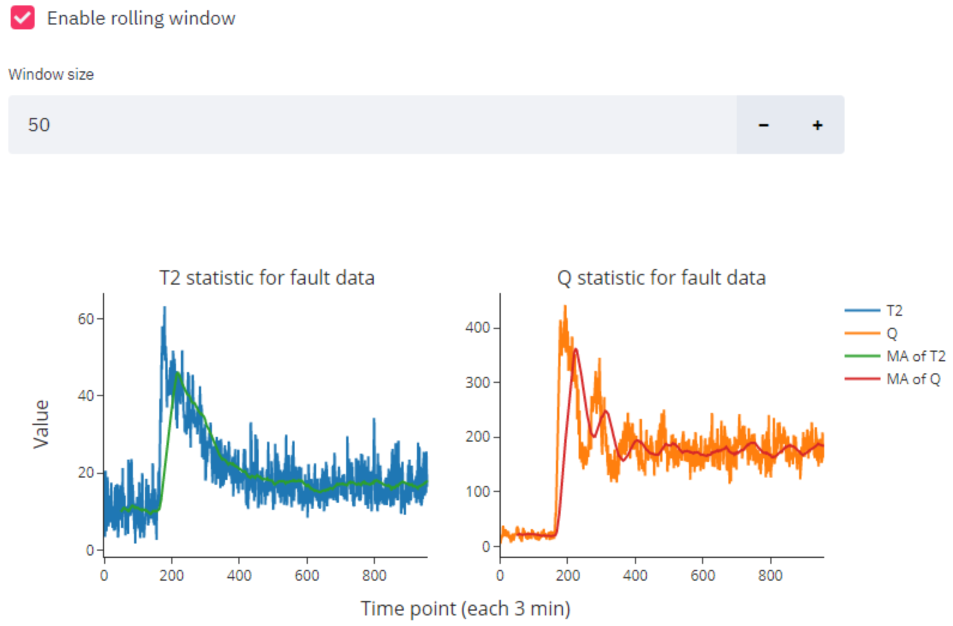


Figure 3. Mean and standard deviation for normalized fault data

The last type of output are the T and Q statistics plots shown in Figure 4. They are also interactive. Basically, the whole analysis is based on them. If the chart is very noisy and it is difficult to determine the trend line, then it is possible to use a moving average on it and select a window for it.


☑ Enable rolling window

Window size

50

Figure 4. Plots of Hotelling and Q statistic

The main application code is contained in the “app.py” file. In order to run it on the command line, in the base directory where the file is located, enter the command “streamlit run app.py”. After a while, the page with the application will automatically open in the browser. If this did not happen, then the local URL link for opening the application will be displayed on the command line.

5. Analysis process

Task is to try to identify following:

- When fault behavior starts?
- Does the process return to normal behavior or stay abnormal?
- Which process variables cause the fault behavior?

Since the task limits us not to use statistics, the definition of the start time will be made on a visual definition. The statistics obtained have a fairly rapid jump-like growth depending on the appearance of a defect. For the purpose of detecting the start time, it is possible to plot the gradient from statistics and from its peak determine the approximate time of the onset of anomalous behavior in the data. For the automatic process of determining the start time, it is necessary to select some kind of threshold, the gradient above which will be considered the beginning of abnormal behavior. This approach requires some prior knowledge of the behavior of the gradient, as well as of the distribution function to determine the limit values beyond the confidence interval. Therefore, we will focus on the most simple, but labor-intensive, visual method of analysis. We will define the start time of the anomalous process as the beginning of the first large jump on the statistics plot.

The determination of whether the process has returned to normal behavior will be based on the nature of the statistics behavior. If the statistics have returned to their original level, then the process can be called returned.

Data analysis methods cannot establish a causal relationship to determine the variable that caused the abnormal behavior, only people familiar with the operation of the industrial process can track these relationships. Analyzing the obtained estimates of the mean values of the time series and standard deviations, it is possible to determine the parameters deviating from the normal behavior. The analysis should always be based on considering both quantities, since, for example, introducing sinusoidal distortions will not shift the mean, but will definitely increase the standard deviation.

Let's take dataset 5, 6, 7 and 9 for analysis.

With automatic determination of parameters for 5 dataset, the following graphs can be obtained, shown in Figure 5. When using automatic PCA and DPCA methods, the result remained the same - the estimated time of the beginning of the anomalous process is 486 minutes after the start of the simulation or 162 points in the data. With hand-based selection of parameters, the plot shown in Figure 6 were obtained. The results obtained do not deviate much from the automatic parameters. Let's take the range for the start time of the anomalous process 155-165 data points.

The curves in Figures 5 and 6 return to their initial level, so we can conclude that the process returns to its normal behavior.

Figure 7 shows the mean and standard deviation for 5 dataset. We will use them to select a number of variables that are out of the general level - these will be our variables with deviations from normal behavior. Selected variables: 0,2,3,6,10,12,15,17,18,19,20,21,24,26,28,30,34,32,37,42,43,45,49,51.

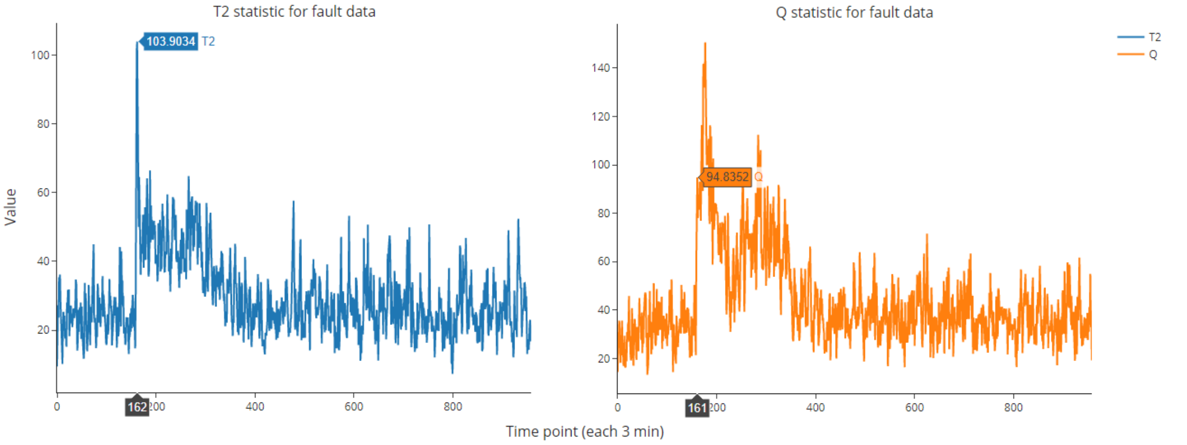


Figure 5. Automated plots for 5 dataset (DPCA, A=28, L=2)

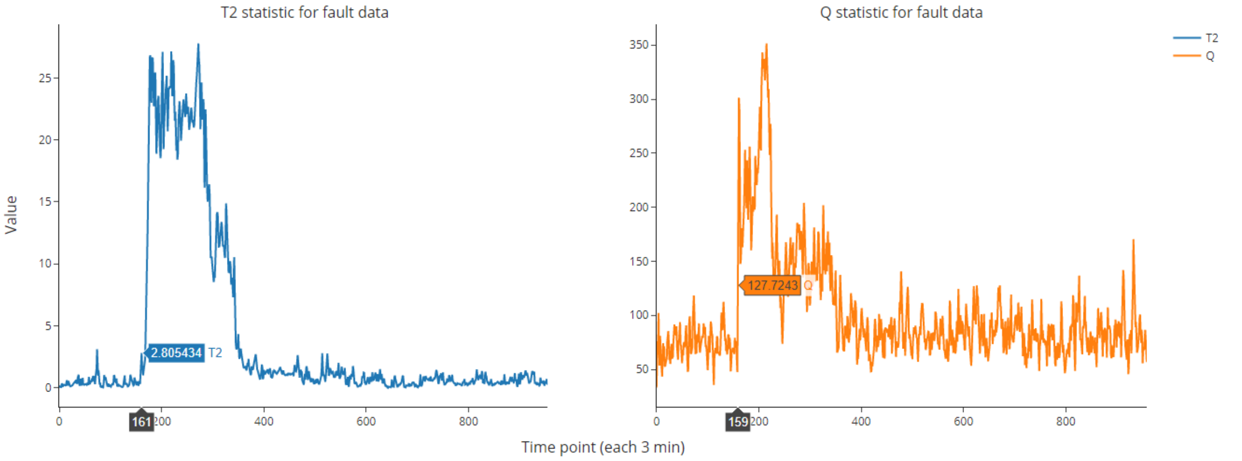


Figure 6. Hand-based approach for 5 dataset (DPCA, A=4, L=2)

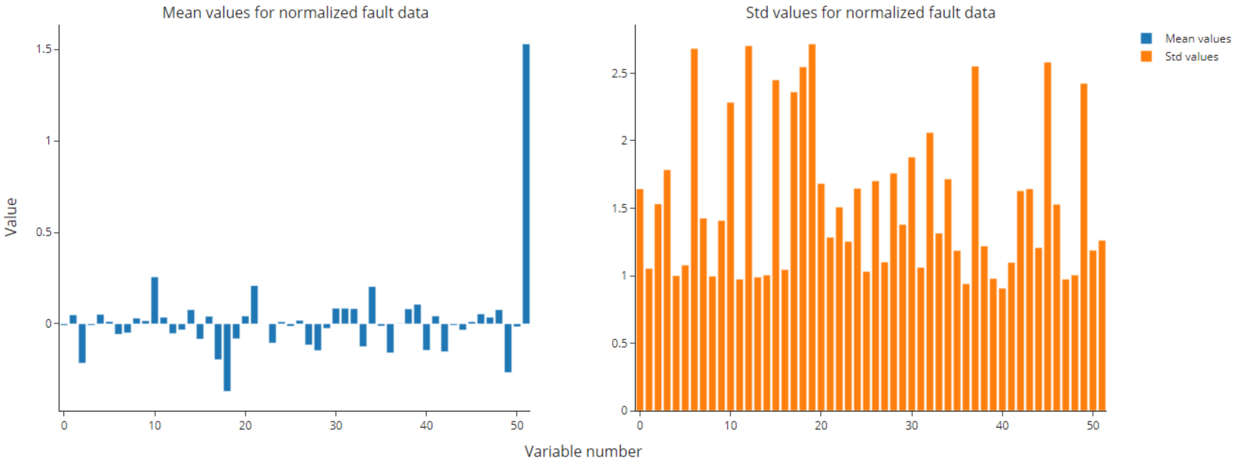


Figure 7. Mean and standard deviation for 5 dataset

For 6 dataset, the plot with automatically selected parameters is shown in Figure 8. For this dataset, the automatic PCA and DPCA methods show approximately the same results. The beginning of the anomalous process is at 474 minutes or 158 data points. With hand-based selection of parameters, the results are shown in Figure 9. As in the previous case, the time of the onset of the anomalous behavior of the system is close to the case of the selected parameters automatically. Let's take the range for the start time of the anomalous process 155-165 data points.

From curves behavior on figures 8 and 9 we can see that curves does not have any tendency to return on normal level. So we can conclude that process stay abnormal.

Figure 10 shows the mean and standard deviation for 6 dataset. We will use them to select a number of variables that are out of the general level - these will be our variables with deviations from normal behavior. Selected variables: 6,10,12,15,18,19,21,22,24,28,30,33,34,35,37,41,42,43,44,45,46,49,50,51.

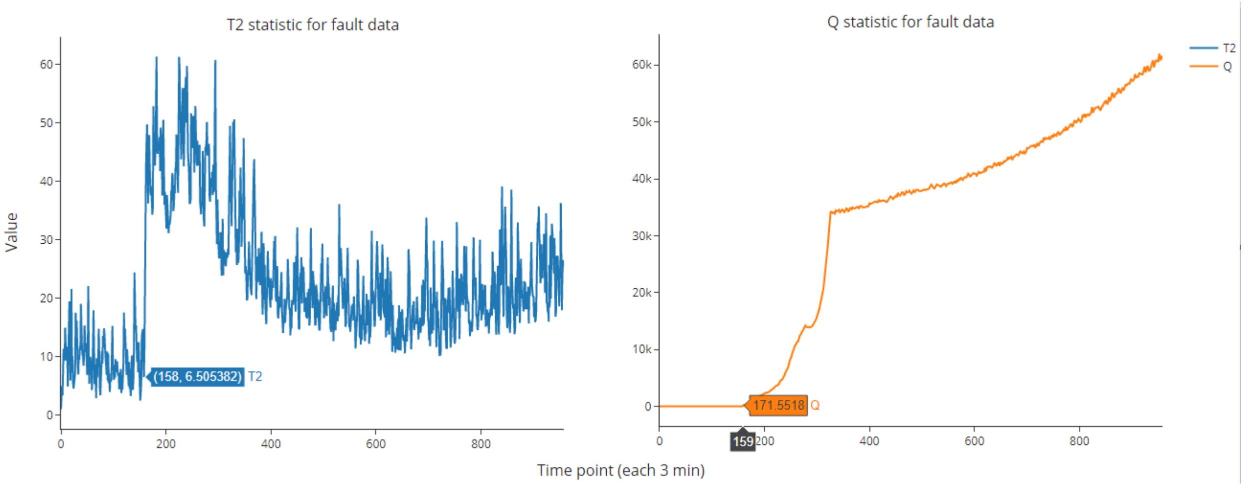


Figure 8. Automated plots for 6 dataset (DPCA, A=17, L=2)

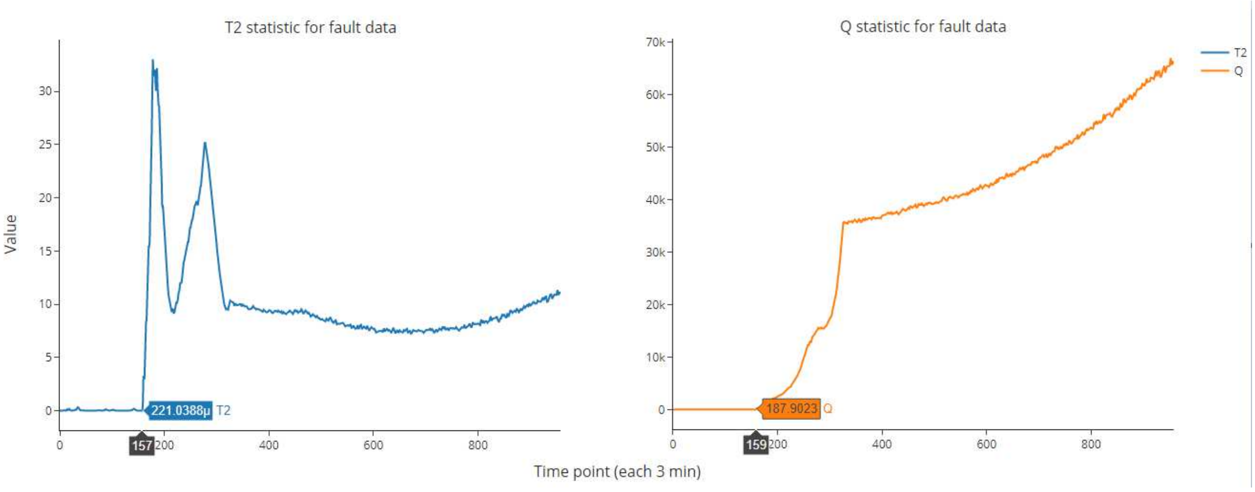


Figure 9. Hand-based approach for 6 dataset (DPCA, A=4, L=2)

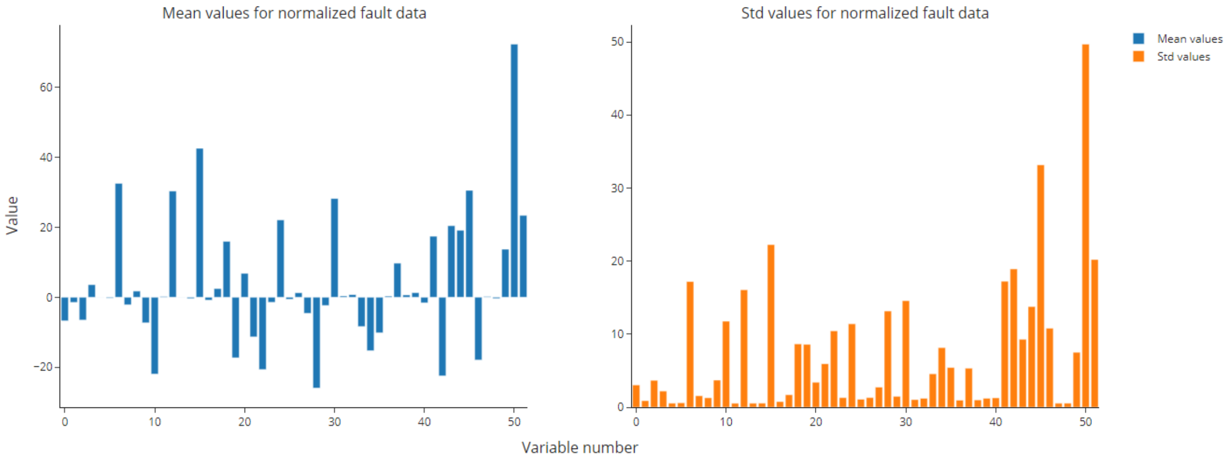


Figure 10. Mean and standard deviation for 6 dataset

For dataset 7 with automatically selected parameters, the results are shown in Figure 11. For this dataset, the automatic PCA and DPCA methods show approximately the same results. The beginning of the anomalous process is at 471 minutes or 157 data points. With hand-based selection of parameters, the results are shown in Figure 12. In this case, a moving average was applied on data with a window size of 10 time points, which means that indeed all points on the plot are shifted 10 points to the left of their original position. Then, to determine the actual time, you need to add 10 time intervals to the selected time point m. Analyzing the obtained result, we can conclude that for the manual method the moment of time 480 minutes after the start of recording was obtained, or 160 data points. Let's take the range for the start time of the anomalous process 155-165 data points.

Figures 11 and 12 show that the time series at the end of the recording remains with abnormal behavior. But if you look at the data yourself, you will notice that most of the perturbed variables returned to their previous levels except for variable 44. Perhaps this is a new steady state for the equipment, but still it is not at the normal level for a dataset without anomalous behavior. We conclude that the process does not return to normal behavior.

Figure 13 shows the mean and standard deviation for 7 dataset. We will use them to select a number of variables that are out of the general level - these will be our variables with deviations from normal behavior. Selected variables: 0,2,3,6,7,10,12,15,17,18,19,20,22,24,26,28,3032,34,37,42,43,44,45,46,49.

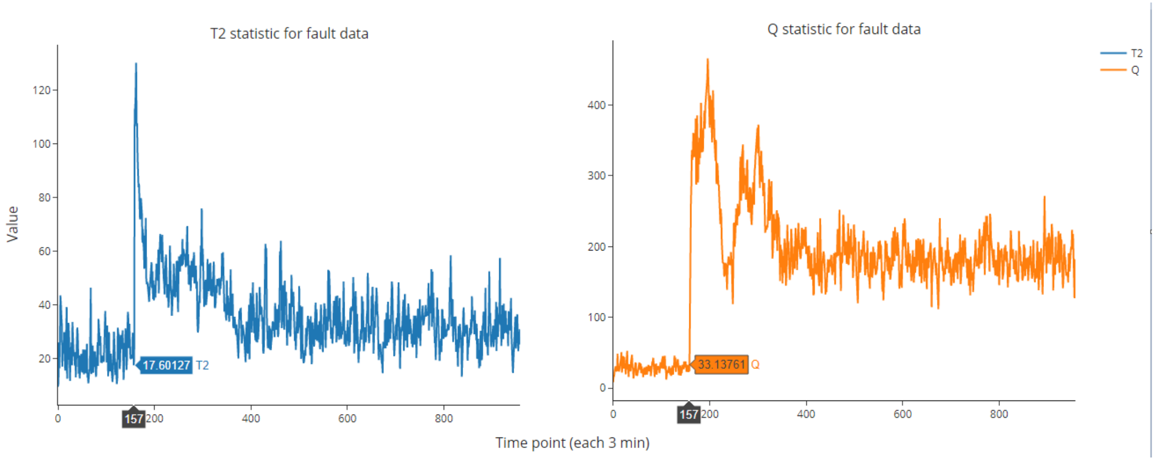


Figure 11. Automated plots for 7 dataset (DPCA, A=30, L=2)

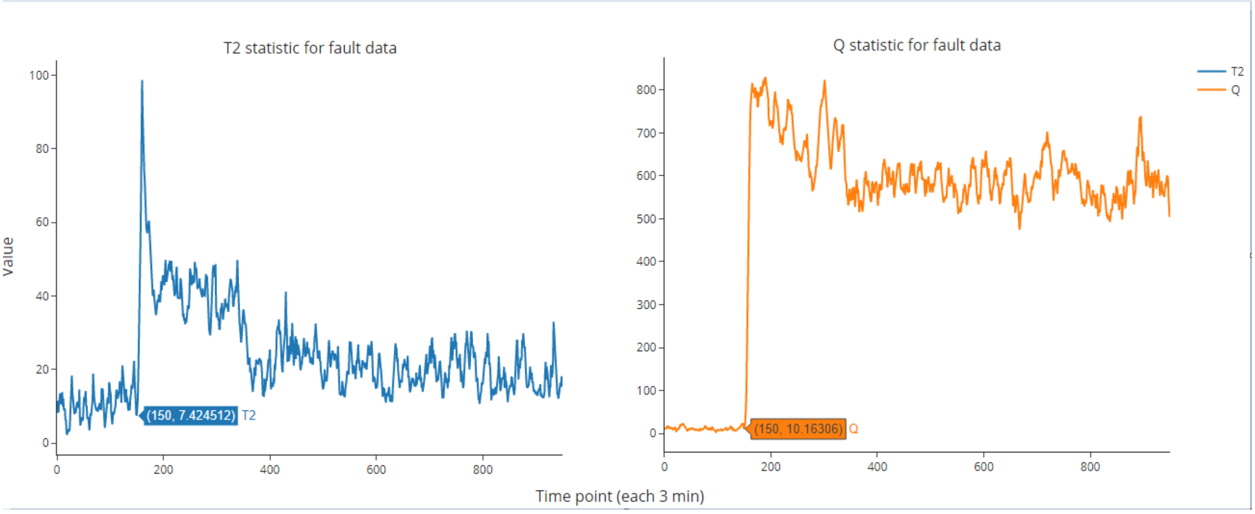


Figure 12. Hand-based approach for 7 dataset (PCA, A=18, window=10)

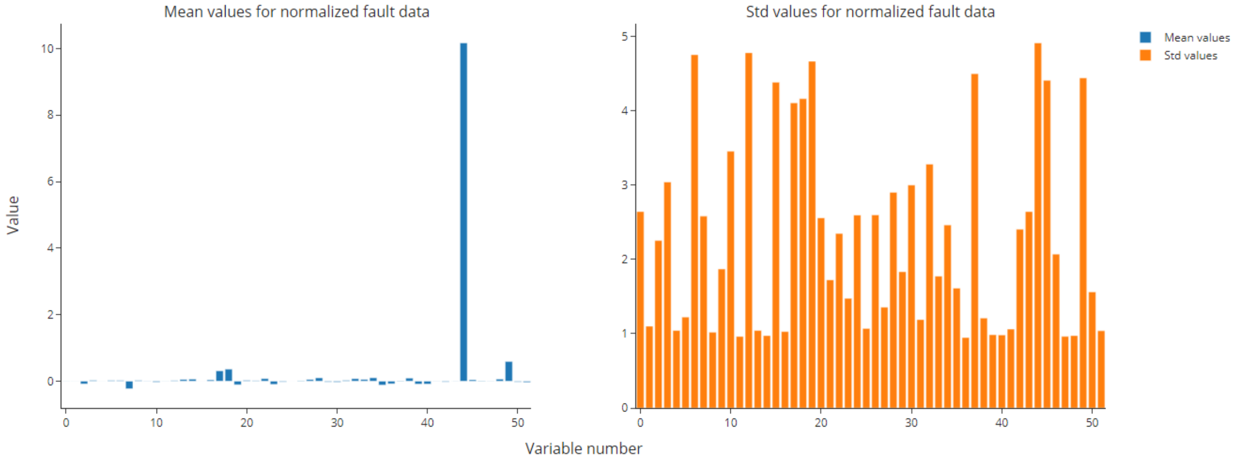


Figure 13. Mean and standard deviation for 7 dataset

It is unfortunate that dataset number 9 does not lend itself to the selected type of analysis, since the resulting PCA and DPCA statistics look like noise (Figure 14), but it is clearly seen that the curves are growing rapidly near zero. We assume that either there is no anomaly in the data, or this anomaly occurs from the very beginning of data recording. If we look at the data ourselves, we will find that many time series are out of phase from the normal data, an example of several found variables is shown in Figure 15. The 9 dataset does not have like the previous datasets jumps in the amplitude of the original signals, so decomposition methods do not work well. With this task, it is possible to get such graphs by running Jupyter Notebook with the name “Fault_detection.ipynb”. Based on the plots, we can conclude that the anomalous process either began shortly at the beginning of the recording or before it began. We will accept the range for the start time of the abnormal process 0-10 data point.

As we can see from figure 15, process did not return to normal behavior.

Since the time series is simply shifted in time, there is no reason to expect that the standard deviation in Figure 16 will show something. But a plot of averages can perfectly show the each time-shifted variables responsible for abnormal behavior. Based on the mean values, we assume that the variables deviating from normal behavior are: 0,1,2,6,10,11,12,14,15,16,18,20,21,24,27,28,30,32,33,34,38,40,47,48,49,51

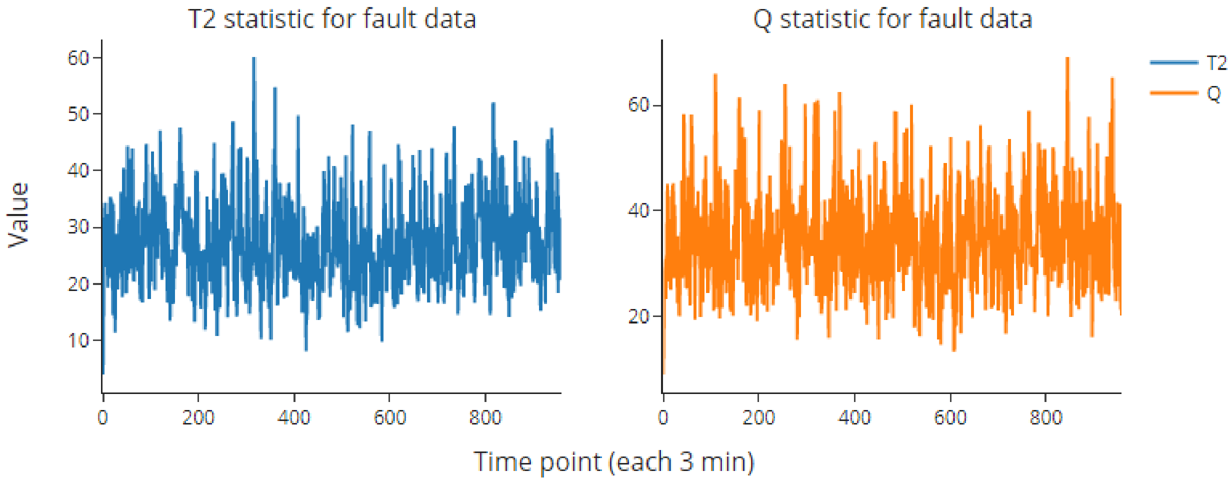


Figure 14. Automated plots for 9 dataset (DPCA, L=2, A=27)

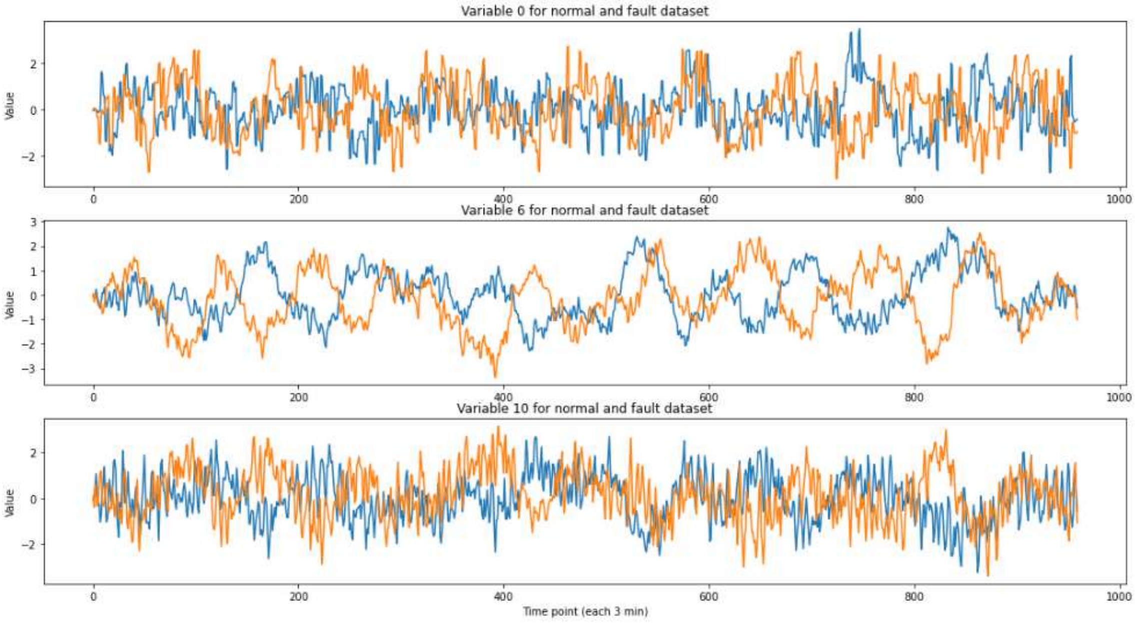


Figure 15. Time series plots for variable 0, 6, 10 from normal dataset and 9 dataset.

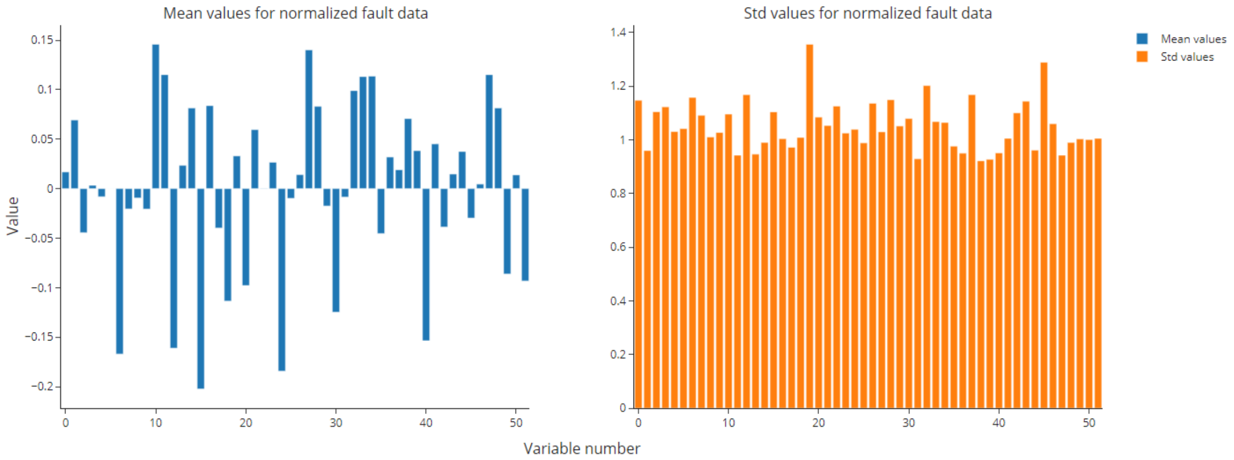


Figure 16. Mean and standard deviation for 9 dataset

6. Conclusion

The paper presents an approach to detecting anomalous behavior in time series based on PCA and DPCA decompositions, as well as T^2 and Q statistics. As a result, the application of this approach gives good enough results to determine the deviation of the industrial process from the normal behavior. The methods make it possible to detect the time of the beginning of the anomalous process quite accurately and using very limited data. However, to automate the process, it is necessary to set boundaries, going beyond which the process will be considered abnormal. The T^2 statistic has a Fisher distribution and, accordingly, it is not difficult to establish the confidence interval. However, as already mentioned, the use of statistical methods requires the fulfillment of certain assumptions about the data, for example, the independence of variables. Obviously, when we receive data from an industrial process that operates as a single industrial unit, the data can’t be independent, and already here the previous assumption is not fulfilled. Therefore, implementing an automatic system for tracking the abnormal behavior of processes, a good step would be to alert the operator of the enterprise about a potential danger, thereby a specialist would be able to check the course of events and make the appropriate decision. Another disadvantage of this approach is that the methods do not help localize the cause of the abnormal behavior, thereby giving no information to the operator about what is causing the detector to “worry”. Nevertheless, this method is excellent for detecting abnormal behavior on a small dataset.

7. Used libraries in code

Short description used libraries in code:

- Numpy - Library for performing operations on vectors and matrices. The SVD decomposition was also used from it,
- Pandas – Library to work with data in table format,
- Sklearn – Library used to perfom PCA decomposition,
- Plotly – Library to create interactive plots different types,
- Os – Library to work with operating system. In our case – find files in folder to analyse,

- Streamlit – Library to create web application. Give nice interactive widget to experiment with data.

8. Bibliography

Evan L. Russell, Leo H. Chiang, Richard D. Braatz, 2000, Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems, Volume 51, Issue 1, Pages 81-93

Pavel Filonov, Fedor Kitashov, Andrey Lavrentyev, 7 Sep 2017, RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process, [Cited 23 Dec 2020]. Available at: https://arxiv.org/abs/1709.02232

Wenfu Ku, Robert H. Storer, Christos Georgakis, 1995 Disturbance detection and isolation by dynamic principal component analysis, Chemometrics and Intelligent Laboratory Systems, Volume 30, Issue 1, Pages 179-196

.