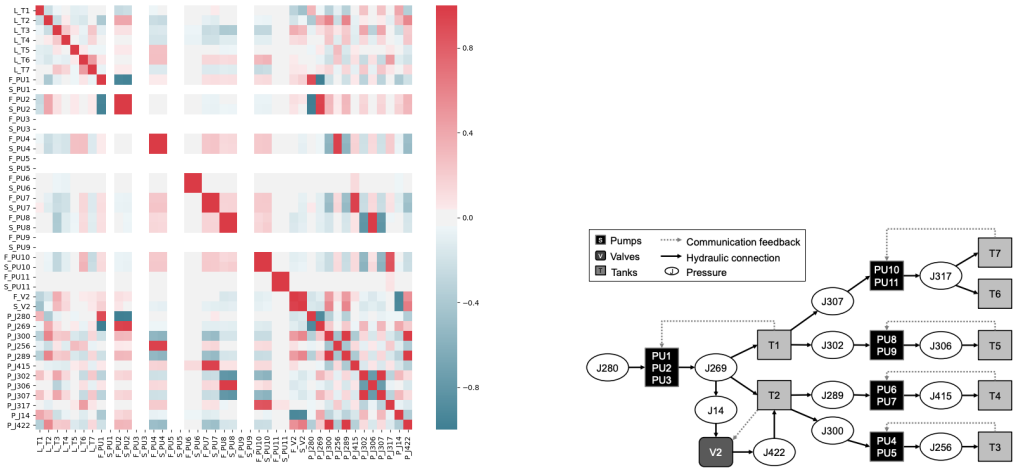
Cyber Data Analytics

Lab 2 Assignment

Kees Fani (4437179) and Jody Liu (4920392)

1 Familiarization task

Signals and correlation: Upon loading the first dataset, the data contains 45 columns with: datetime, water levels of tanks, binary on/off actuators for the pumps and valves with its flow level and 11 suction pressure and discharge pressure for each pump and valve. Lastly, the attack flag indicates if a row is an attack. Based on the BATADAL paper and the above features there are thus 43 signals; with actuators and sensors. To evaluate the correlation of the signals we created a correlation matrix for the 43 signals; excluding `datetime` and `attack flag`. From Figure 4 there are signals with strong correlation shown in dark red/blue color. Comparing this to Figure 4b it shows the correlation heavily depends on the signal placement. E.g. P_J317 has only a high correlation with pump 10. This is reasonable because they are connected with one another based on Figure 4b.

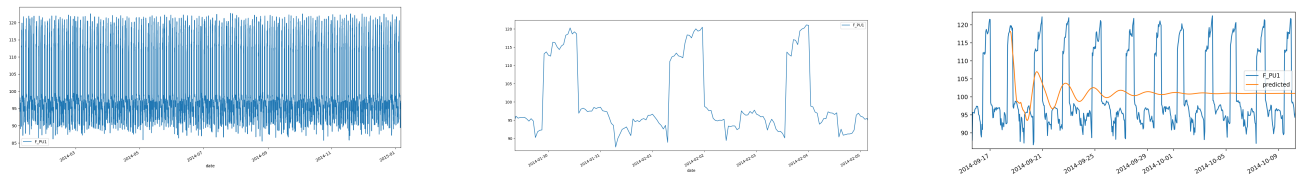


(a) Correlation matrix of all the signals

(b) Physical connections of the signals

Figure 1: Correlation matrix and the visual of the physical connections between the signals

Cyclical behaviour and first initial predictions: To evaluate whether the data contains cyclical behaviour we took a signal that correlates the most and plotted the signal over time. With the build-in zoom function of Matplotlib we took different snapshots of the first pump (F_PU1) as signal since the first initial Figure 2a is too dense to draw conclusions. Figure 2b shows there is indeed a cyclical behaviour. A clear peak occurs followed by a dip on the second day, a small increase on the third day followed by a dip, where again a new cycle occurs with a high peak. With this observation we can perform a first prediction based on the principle of auto-regression mentioned in Lecture 4. Figure 2c illustrates an initial, simple prediction by using an Auto Regressive model. An AR-model is a regression of the future values based on the historic values. Since this is an initial model to see if signals in the data can be predicted, we did not perform any parameter tuning. Again, with the Matplotlib-tool we zoomed-in to evaluate the results better. Based on Figure 2 the AR-model is able to predict the first initial future values in 09/18/2014 quite well. In other words, the next future value based on the historic value is quite easy to predict. But as we go farther in the future it becomes harder to make such a prediction; resulting in an almost straight horizontal line.



(a) Complete plot of the dataset

(b) Zoomed-in plot of a week

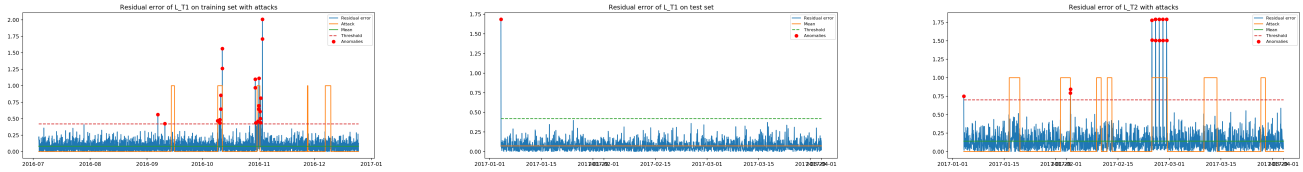
(c) First prediction test using AR

Figure 2: Pressure level of pump 1 over time

2 ARMA

NOTE: the choice for the 5 sensors, its preprocessing and main findings are explained in Section 5. Similar to the sections about the discrete models and PCA, this section only focuses on the model and how it is build.

We used (partial) autocorrelation to determine the maximum values for p and q of the ARMA model. The last point that falls above the confidence interval was chosen as maximum. With the recommended Akaike's Information Criterion, we used a search function on the ARMA model to find optimal parameter values. If the AIC does not reduce any further, the function stops and chooses the corresponding terms. This was done on the first training set. We then used the second set for validation and optimization: 1) we noticed that one (partial) autocorrelation of one signal cannot determine maximum p and q for all signals; the prediction models did not fit well for the other 4 signals. We therefore ran for each signal a separate (partial) autocorrelation. 2) we used the residual error *(prediction - actual)* to determine the anomalies. As the process managers in the BATADAL paper mentioned: they manually decided on a threshold for each signal. We thus used the validation set to determine our own threshold as well. This was based on how the attack labels were positioned where we focus on point anomalies for abnormal peaks (Figure 3a) and collective anomalies with unusual sequential patterns (Figure 3c). Figure 3a shows that the ARMA model finds most anomalies corresponding to its signal T1, even the ones BATADAL has not found (= the first red dots in 3a). Also the residual mean is close to 0, meaning it fits well on the normal data set. 3b shows a detected anomaly in the test set. However, the ARMA model does not optimally work for example T2: it did not detect all anomalies. This can be explained from possible noise in the data and the manual threshold that is set.



(a) Validation set on T1 (b) Finding new anomalies on test set for T1 (c) Test set on T2

Figure 3: ARMA model on tank 1 and tank 2 as signals

3 Discrete models

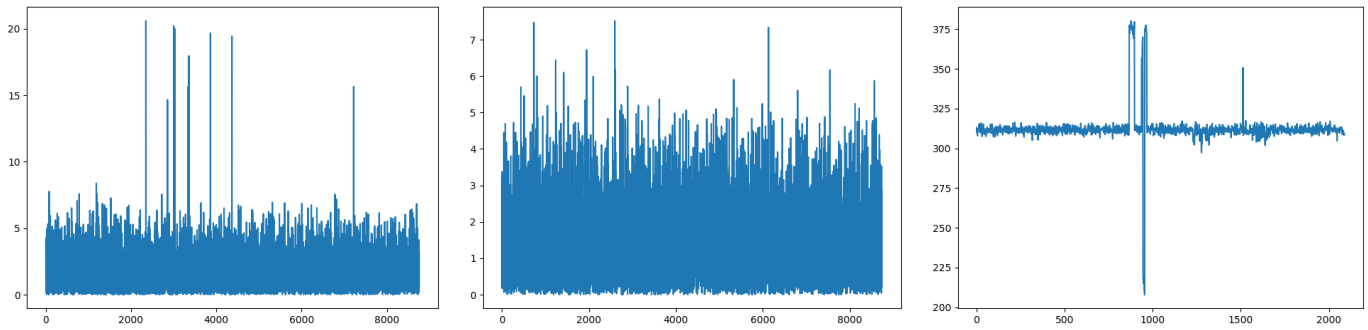
The sensor data was discretized (e.g. converted to accba, bbcca etc.) using the sliding window SAX algorithm. An empirical analysis was used to determine the optimal parameters for the discrete values to successfully find most anomalies in the given signals. The essence with this model is to find patterns of the discrete values in the test signal that do not occur in the training data. Since these patterns are discretized, this check can be performed using a simple mapping from pattern to location. These locations can then be used to localize the detected anomalies. Here we observed that the higher the window size was set the more false anomalies were detected. We mainly found point anomalies because of abnormal sequence patterns found compared to the normal data (Table 1).

L_T3	L_T2	F_PU1	L_T1	L_T6
2017-01-18	2017-01-05	2017-02-12	2017-01-18	2017-02-10
2017-01-18	2017-01-31	2017-02-25	2017-01-18	2017-03-26
2017-01-18	2017-01-31		2017-01-30	2017-03-26
2017-01-19	2017-02-10			
2017-03-26	2017-02-10			
	2017-02-23			
	2017-02-24			
	2017-02-25			
	2017-02-26			
	2017-02-27			

Table 1: Detection of anomalous signal sequences on test set. Red means that the anomaly was a false positive, yellow means that it was off by a day and green means that the anomaly detection was correct.

4 PCA

The PCA anomaly detection analysis was performed in three steps. First PCA was performed on the training data to obtain a fitted PCA model where we used an explained variance ratio to determine the number of principle components. In this case that was 10. This allowed us to compute predictions of the signal data. Residuals were calculated by taking the absolute residual *(training data - predictions)*. This showed unusual abnormalities in the data that should not be present. The peaks (> 10) in these residuals were therefore filtered out of the training data to create a baseline. This new data was then used to fit the model. Lastly the test data was transformed using the PCA model to obtain a prediction of the test data. This prediction was used to calculate the residuals of the test data.
Using a threshold of 325, we can see that the two biggest peaks correspond to attacks #9 and #10 of the test data. The third peak is a false positive. This approach finds point anomalies because it looks at patterns independent from the context of the data.



(a) Residuals of the unfiltered validation data (b) Residuals of the filtered training data (c) Residuals of the test data

Figure 4: Residuals of the different stages in the PCA analysis

5 Comparison

Data - For the three model types we used the signals: L_T1, F_PU1, L_T2, L_T3 and L_T6. The choice for the first two signals is because they appear in the second training set. This is important to evaluate how well the model performs based on the validation set and is used to optimize the model with parameter tuning. The other three signals are used to evaluate how well the models perform on the test set. For all models we first applied a normalization to remove the mean variance for a better baseline model. This creates a higher variance between the baseline and the anomalies.

Evaluation - There were two metrics that we considered: the Time till Detection (TTD) (Figure 5) and the confusion matrix. Aside from detecting each anomaly correctly with a high true Positives and low false Positives, it is also important that the anomaly is detected in time. If the detection is too late, then it might be too late to prevent the high negative impact of an attack. The drawback of the formula is however the inability to detect false negatives. In other words, for each detected anomaly it will calculate the TTD. This makes the metric misleading if the models are highly inaccurate. Because both metrics are in this concept contradictory where one takes false negatives into account and the other not, we decided to use a confusion matrix where we deem it important to detect each single true positives and negatives.

Result - $\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$ $ARMA$ $\begin{bmatrix} 1674 & 3 \\ 397 & 15 \end{bmatrix}$ $Discrete$ $\begin{bmatrix} 1671 & 6 \\ 396 & 16 \end{bmatrix}$ PCA $\begin{bmatrix} 1675 & 2 \\ 362 & 50 \end{bmatrix}$

We noticed in most cases the models found true anomalies, but also only parts of the complete anomaly range (second attack in Figure 3c). This is due to possible noise in the data but also the signal levels that feel the effect of an attack later. In this case the model will understandably not detect such an attack immediately and this can thus in part explain the large number of FN's for all three matrices. While the PCA appear to perform better it is more prone to outliers as we had to refit the model to eliminate this caveat. The ARMA and Discrete model used manual thresholds and could explain the lower performance. However, the ARMA is recommended for more flexibility and is visually easier to analyse to create more robust models. On the other hand, if the user prefers runtime then we recommend the discrete model as it lower the continuous values into discrete ones; which reduces the data size and is able to perform even better than the ARMA. In our case we prefer the ARMA because it is visually easier to interpret the data over the timeline where certain peaks and sequential variations are better captured in context.

$$S_{TTD} = 1 - \frac{1}{n_a} \sum_{i}^{n_a} \frac{TTD_i}{\Delta t_i}$$

Figure 5: Formula of the Time till Detection based on the BATADAL paper