Cyber Data Analytics (201500037)
Assignment 2

Group 42

August 9, 2018

1 Data analysis and prediction

We started the assignment by loading the training data 1 and plotting the values of each sensor in order to familiarize with the data. The values contained in the datasets corresponds to signals hourly produced by the monitoring sensors in a simulated water distribution system. Those signals corresponds for example to the level of water in the tanks.

For each pair of sensors we calculated Pearson's coefficient on the normalized data in order to estimate to what degree they are correlated, it turns out that the values recorder by some sensors seems to be highly correlated with some signals of other sensors.

We can easily spot this correlation by the mean of a simple plot. As an example let us have a look at the first 100 occurrences of columns P_J300 and L_T2 as shown in Figure 1:

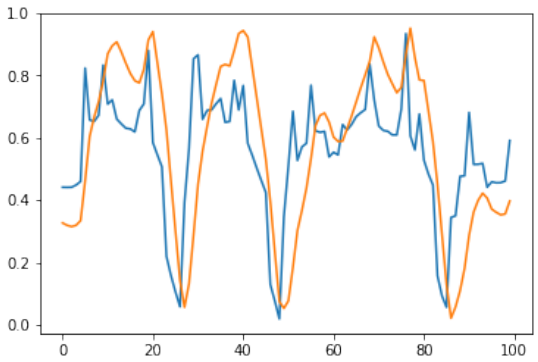


Figure 1: First 100 values of columns P_J300 (blue) and L_T2 (orange)

In addition we noticed that the signals often show a cyclic behavior (as shown for example in Figure 2).

We were able to predict the next value in a sequence of signals using Markov chains, Figure 3 shows the predictions of our model against the real (discretized) values on the datasets. We will discuss more in depth of this approach on the next section

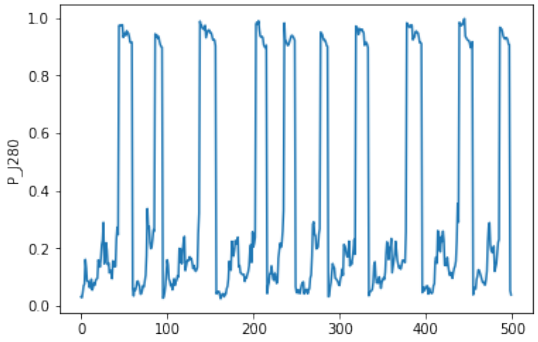


Figure 2: Example of cyclic behavior of a signal (signal P_J280)

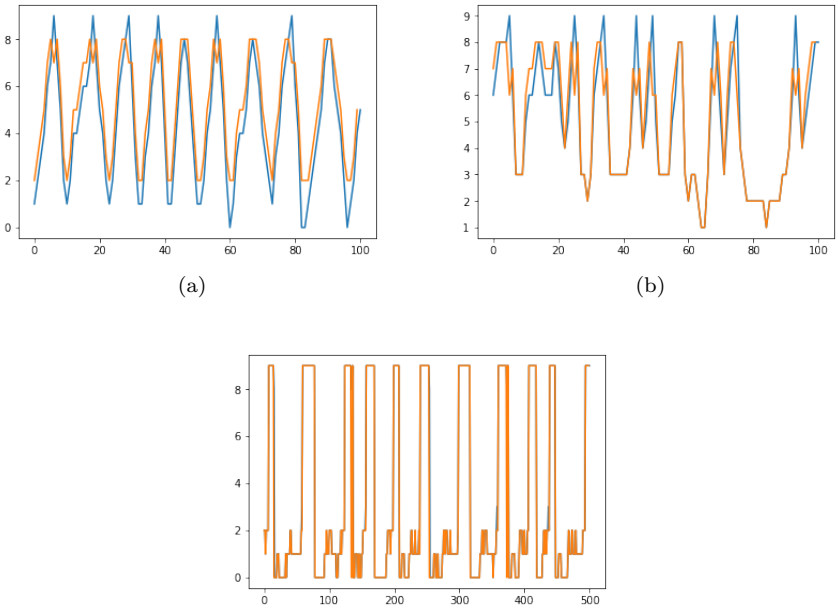


Figure 3: Predictions of the discretized signals L_T5, L_T7 and P_J280 (respectively (a), (b) and (c)) using Markov chains (the original value are in blue, the predicted values are shown in orange, as shown by the y-axis the signals are discretized over 10 points).

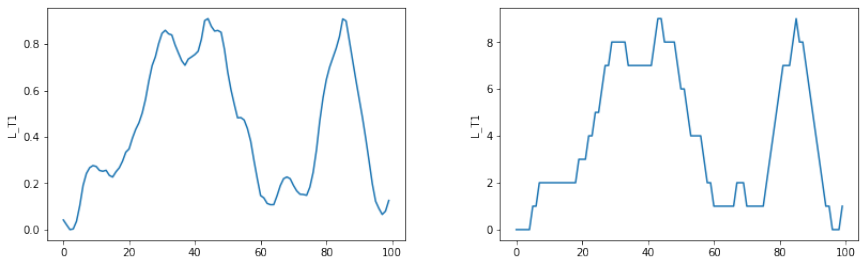
2 Anomaly detection

Auto-Regressive Moving Average models are ideal for predicting values on time-series data. For this part of the exercise we have trained an ARMA model, based on the statsmodel library, for each signal with a non-0 standard deviation. We experimented using autocorrelation plots for finding the optimal number of lag observations(p) for the models but, in the end, we decided to use grid search. By using Akaikes Information Criterion (AIC) as our reference, we could find the most optimal lag observations and moving average order pair that would yield the most accurate results for each model. In fact, the lower the AIC for a given set of parameters, the better the fit on the dataset. Having the optimal parameters for each signal, we trained an ARMA model for each sensor on the first dataset and then made predictions for the signal values on the second one. After each prediction, the models were trained over to include the previous prediction.

Having a prediction and the original signal value for a given timestamp, we are able to determine whether there is an anomaly by deducing the one signal from the other and comparing that delta value with a predefined threshold.

For this part of the assignment we first discretized the dataset using symbolic approximation of time series via equal width binning and then used Markov chains to try to detect anomalies in the dataset. We decided to use equal width binning as a discretization algorithm for its ease of use and relatively small number of states necessary to model the data without any excessive loss (in addition the reduced number of states makes it possible to easily visualize the transitions used in the Markov model).

Figure 4 shows a signal before and after discretization.



(a) (b)

Figure 4: Signal L_T1 before (a) and after (b) discretization using equal-width binning with 10 bins

In order to detect anomalies we used the training set to create a Markov model for each feature in the dataset and then applied this lasts to the other datasets using a sliding windows of length 80 (this was the length that turned out to perform better). We then set a threshold for each feature and raised an alarm each time the probability of a certain sequence was smaller than the given threshold. By counting the number of alarms raised among all features we were able to detect anomalies in the system. Figure 5 shows how, by setting the threshold of raised alarms at 9 (i.e. 10 or more alarms imply an anomaly) we were able to detect all the attacks in dataset 2.

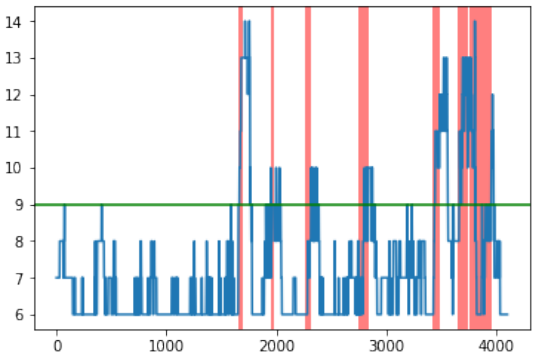


Figure 5: Detection of anomalies in dataset 2. The zones in red correspond to the attacks.