**1b**

SCADA is a system/network of software and hardware components in which the data from sensors and other industrial components like pumps and valves can be monitored and controlled from a centralized computer. These kind of systems are vunerable to cyber-physical attacks.

The BATADAL dataset is a representation of (measured) data over time from tanks (7), pumps (11), valves (4, of which 1 actionable), and other hydraulic components like joints (12 in total) from the C-Town water distribution system. The information that is embedded in the dataset contains the water level in each tank, the flow through each pump and valve and their status at that time (0 for off, 1 for on), and the pressure in the joints. Each record additionally stored the date and hour, and the attack flag status (0 for no attack, 1 for attack, and -999 as unknown(?)).

##### Signal types

The signals that can be extracted from the dataset thus are the levels in the water tank, the flow through the pumps and valves, the status of each pump/valve over time, and the pressure in the joints. In Figure 1 all four signals can be seen over about a month in time for illustration purpose only. We can clearly see that the patterns for all signals are between boundaries, and look like to be repeating/cyclic in some form. We also see that the status indeed switches from off and on, although the value is only measured once every hour.

##### Correlation

To see if the signals are correlated we can determine the correlation between any columns in the dataset, and create a heatmap to easily visualize the correlation (see Figure 2). We see that most signals have a no correlation (value +-0), while there exist some perfect correlated signals (excluding the signals with theirselves) and perfect negatively correlated signals (signals move in opposite direction). For example the correlation of the flow of pump 2 (F\_PU2) with the status of pump 2 (S\_PU2) is 0.998614, meaning they are perfectly correlated. This is ofcourse not strange since if the pump is enabled (status code 1) the flow in the pump is also enabled, see Figure 3 for pattern comparison of a part of the signals. If we look at other signals (that are not from the same component), we see that for instance the flow in pump 1 is almost perfectly correlated with the pressure in joint 280, with a correlation value of 0.907052. This probably indicates that joint 280 is the connection leaving pump 2, while it is connected to something that is not flowing at the same speed as pump 2. We also see (Figure 4) that for instance the signals from F\_PU1 and F\_PU2 are perfectly negatively correlated with a value of -0.949012, meaning that when the signal from one of the two is going up or down, the other one is moving in the opposite direction.

From the signals we can easily see that they look to be cyclic. From the flow in pump 2 signals in Figure 3 we see that in 24 hours (although it is measured ones per hour) the frequency of the signal repeating is about 14 times. The cycle is not a perfect sinusoidal, but the similarity between them is very well visible.

###### Prediction

For the prediction of the signals we use an AutoRegressive model that is trained by a training set, and for the prediction it takes previous observation(s) of the time series into account. The lag has been chosen to be closer to zero and not bigger since the more recent previous observations should be more important than extra previous observations (adding more lag decreases the influence of all other observations). In Figure 5 the original and predicted signals of the water tanks can be seen, including their RMSE. We can see that for all signals did well, although for signals L\_T4, L\_T5 and L\_T7 the prediction is less accurate than for the others. If we compare signals L\_T1 and L\_T6, at first sight L\_T1 looks like to be predicted better by a lot. If we look at the RMSE the difference is just a fraction.

If instead of the water levels of the tanks we try to predict the signals of the flows of the 11 pumps, the result is very different. The average RMSE is increased to 7.59, and the predicted signal does not follow the curves of the original signal at all for most signals.

Predicting the next value in a series thus differs if the signal is applicable for it. From what we've seen is that the water levels in the tanks are very fitting for prediction, while the flow in the pumps are not.

**2b**

**3b**

Before we can apply PCA we must normalize the training dataset 1. Since PCA is affected by scale the features must be scaled before it can be applied to it. To obtain optimal performance in machine learning algorithms it requires the features of the dataset to be unit scale, with zero mean and variance one. We remove features that contains NaN values. The remaining 36 features can be used for PCA decomposition. For PCA to work with anomaly detection it must lower the number of components because otherwise it has a difficult time differentiating anomalies from normal cases. The cumulative variance of the principal components (n) determines the number of components for a defined threshold. To be able to detect the normal behaviour of the series, enough variance must still be present. However, too much variance can mean that the model has learned about the anomalies as well.

In Figure 3.1 the PCA residuals of all signals with the corresponding number of components can be seen. It looks like that 99% removes too much anomalies, while for lower than 97% it is hard to distinquish what is an anomaly and what not. The total error increases a lot if the variance is increased. We therefore choose to use a cumulative variance of 97%, containing only 13 components. In Figure 3.2 the number of components is displayed against the cumulative variance. The last 21 components add a lot of error to the residuals, and are considered to be of less use.

The next step is to remove anomalies from the dataset. To remove the outliers we have to set a threshold. We assume a normal distribution. The threshold for normal behaviour samples is estimated to be mean plus/minus three times the standard deviation [1] for each signal separately in the original dataset (after processing). For this interval there are 85 anomalies detected and removed. Some large anomalies are present in the dataset, see Figure 3.4. All signals now behave in somewhat the same boundaries. The model should be trained after these anomalies are removed as much as possible, otherwise the model also learnes information about the anomalies thinking it is normal data. In some situations it may occur that the values of the sensors in the SCADA system do not follow the correct patterns due to for instance an operator that is evaluating a problem, a broken sensor, or a software bug.

The performance of the PCA-based anomaly detection can be evaluated by applying training set 2 to the trained model. We search for a suited threshold on the residual error, in which most attacks can be recognized by just looking at the residuals. The threshold is found by optmizing the precision (maximizing TP and minimizing FP). Training set 2 contains attack labels, such that we can easily evaluate the performance. In Figure 3.5 the residual plot of training set with the calculated threshold is visualized, in combination with the attack labels. It has an accuracy of 91% and a precision of about 96%. There are only 5 false positives, compared to the 123 true positives, which shows good performance for the anomaly detection.

We can also look at the anomaly detections in detail in Figure 3.6, in which a couple of situations are visualised. The first plot indicates a point anomaly and the second one a collective anomaly. There is no contextual anomaly detected.

[1] VARUN CHANDOLA, ARINDAM BANERJEE, and VIPIN KUMAR. Anomaly Detection: A Survey

**4c**

Before we can apply ARMA to the signals of the sensors we have to pre process them. We remove the columns that are not interesting for finding anomalies: binary values, mean zero, variance zero. We could also determine whether the signals of the sensor we will use are stationary, but since ARIMA (instead of ARMA) can deal with stationary, we will not research this. We first have to figure out the order of the ARMA models, by applying the Akaike's Information Criterion in a grid search to figure out which parameters suit the model best given the training data. We can also look at the autocorrelation and partial autocorrelation plots to verify the numbers.

The only sensors in the SCADA system are the ones measuring the water levels in the tanks (tank 1 to 7). For each of the 7 sensors we are going to train the model, using the computed parameters p and q, based on the training data 1, and predict the data based on the test data. The test data contains attack flags, which we can use to evaluate the performance. The residual error is computed by taking the absolute values of the difference between the values of the test set and the predicted values. The anomalies are then the values that are bigger than a defined threshold. The threshold is estimated to be equal to the mean plus three times the standard deviation of the residual error, which in total includes 99.7% of the points to be 'normal'. We plotted the figures containing the predicted anomalies, residual error, threshold and true attacks in one figure for each sensor to show the result. The metrics are also given for each sensor. We see that the overall accuracy is quite well, although the precision and recall aren't that well. The anomalies that are detected are point anomalies and collective anomalies. These types are suitable for the threshold-based detection. If we look at the results of the anomalie detection of each sensor, we can clearly see that most are not suitable to be modeled using ARMA. There are two sensors that look like to perform relatively fine compared to the others: L\_T2 and L\_T3. In most cases there are more FalsePositives than TruePositives, and a lot of FalseNegatives and TrueNegatives.