Detection of Attacks on a Water Treatment Unit

Robin Khatri, M1 MLDM

GitHub Repository: All codes and visualizations along with some helpful documentation are present at `https://github.com/robinredX/Data-Mining-Project`

1 Introduction

Water treatment units, power grids, nuclear plants and other types of industrial infrastructure form backbone of infrastructure. These units usually operate on SCADA (Supervisory Control And Data Acquisition) [2], and are increasingly becoming the target of cyber-attacks. In 2011, a cyber-attack focused on simply shutting down a water pump in Illinois [3], while in 2016, hackers were able to change the levels of chemicals of a drinking water process [13]. These are just a couple examples of a highly vulnerable domain.

In this project, we intend to detect whether a water treatment system has been compromised, based on analysis of observations from its physical properties. For this purpose, we used Secure Water Treatment (SWaT) Dataset, that was acquired from the iTrust Laboratory of Singapore University of Technology and Design [5].

The dataset is sufficiently large and has many features underlying different physical processes and attracted me due to my interest in cyber security.

2 The Dataset

2.1 SWaT testbed: Processes

In 2015, the iTrust lab of the Singapore University of Design and Design built a mini-scale water treatment unit (the SWaT testbed, or the Secure Water Treatment unit), where all the components of the physical process were connected online, and by doing so, the system could be tested and observed for incidents of cyber-attacks [5]. It is fully operational SCADA based industrial control system. [9] [5].
A full overview of the processes in the various stages can be seen in Figure 1 [5]. Previous studies on the dataset has focused on identification of types of attacks and behavior of those attacks. [6] [7] [8]. In this project, focus is to understand the underlying processes, and to establish relationships among them to help in selecting features that are suitable for detecting the attacks accurately with the help of data mining and machine learning methods.

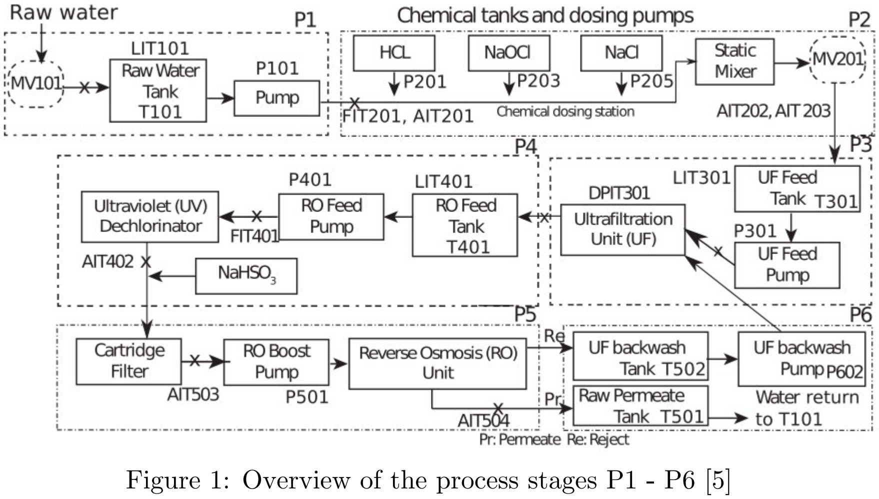


Figure 1: Overview of the process stages P1 - P6 [5]

2.2 SWaT testbed: Data

During the data collection process, the researchers run the testbed for a total of 11 days. During the first 7 days, the system operated normally, while the remaining days were used to launch various attacks.

Data files:

SWaT Dataset Attack v0.xlsx (113 MB),
SWaT Dataset Normal v0.xlsx (127.8 MB)

Basic information about the dataset:

Number of instances/observations: 946,719
Number of features: 53

On official dataset, there was a third data file that had the same observations as the Normal v0 file except the instances during the first 30 minutes. The first 30 minutes, the tank was empty but still there were readings because of the continuing maintenance of the unit. We therefore did not used that file. More information on the process of launching attacks and dataset preparation can be found on the official documentation for the data `https://itrust.sutd.edu.sg/research/dataset/dataset_characteristics/#swat` [5].

3 Data understanding and pre-processing

Each feature is a reading from a sensor or an actuator [9] [5]. By Figure 1, you can view what each feature represent. Now we'll explore these features and get better understanding of the data and prepare the datasets for modelling. After data import to R, each feature is classified as numeric. We'll analyse further on whether all features are continuous or some features are in fact categorical.

3.1 Data Quality

First we checked for missing data points in the dataset.

```
> which(is.na(dataset)) # Check for missing data (NA)
integer(0)
```

There was no missing data. We'll also look into Timestamps which is the first feature of our dataset. We shall not use this feature during our modelling since the attacks were carried out at pre-planned times and fitting a prediction for attacks with respect to time will be misleading, however to understand the distribution of our events (attack or normal) and to filter our dataset for modelling, it is a useful tool.

```
> convert_to_dateobj <- function(dataset$Timestamp) # We created this function to convert
timestamp into the date object as it was a character originally
[1] "Success"
```

```
> which(is.na(dataset)) # Check for missing times
integer(0) # There is no missing data
```

```
> time_vector <- seq(ymd_hms(starttime),ymd_hms(endtime), by = '1 sec')
> missing_times <- time_vector[!time_vector %in% dataset$Timestamp]
> range(missing_times) # There is one missing period
[1] "2015-12-31 21:00:37 UTC" "2015-12-31 21:01:57 UTC"
```

So, in the dataset, 81 seconds of observations are missing. Due to this, we need to check if this missing-ness is at random or it influences the values of other features. For this purpose we plotted two graphs over a duration of 1 hour for every feature in our dataset - one before the discontinuity of timestamps, and second after it. These plots did not show any influence of missing timestamps on our dataset. For example, figure 2 represent readings of FIT101 before and after timestamps went missing. Other features show similar evidence. So, we can be sure that nothing dramatic happened during the times for which we do not have data.

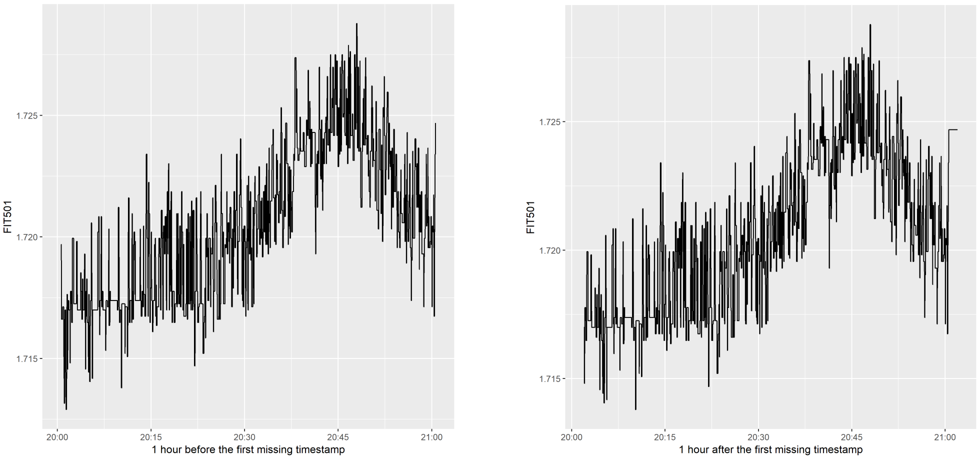


Figure 2: 1 hour before (left) and after (right) timestamps went missing

3.2 Target variable

The target variable in our dataset was ‘Normal/Attack‘ and it was binary in nature taking on two values - Normal and Attack. These values were replaced with 0 and 1 for analysis purpose.

There are in total 892,098 (94.23%) normal instances and 54621 (5.77%) attack instances. The dataset is unbalanced. We’ll keep this in mind during training our modelling.

3.2.1 Distribution of target variable over days and hours

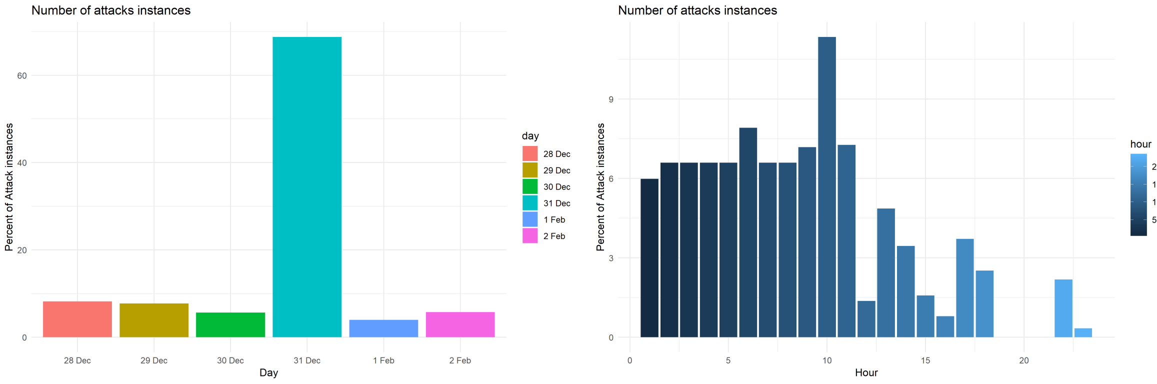


Figure 3: Distribution of attacks by date (left) and hour (right)

3.3 Categorical features

From summary of the dataset, it was clear that some features are in fact categorical. We created a function to identify binary variables.

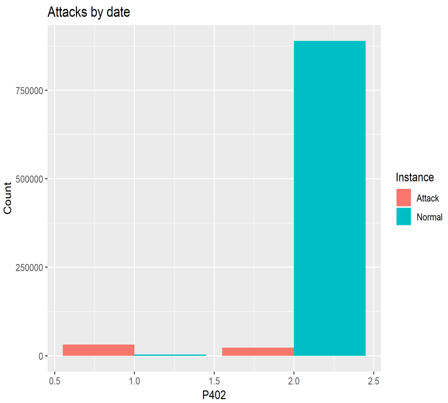
```
> cat_indices <- find_cat(dataset) # Returns indices of features that are binary
> cat_indices
[1]  5  6 12 13 14 15 16 17 25 26 31 32 33 34 35 44 45 50 51 52

> zero_var_indices <- findvarzero(dataset) # Returns features that have 0 variance
> zero_var_indices
[1] 13 31 34 45 50 52
```

After finding our categorical variables, we wanted to see the distribution of these features for both classes. We found that some features had only one value i.e. they do not change at all. We defined a function for this purpose.

We plotted distributions for all categorical features. For binary features P402, P501, UV401, one category is significant during attacks. We used this information during feature selection as detailed later. Distribution of P402 for example is presented in figure 4.

Figure 4: P402: One value is significant only during attacks



3.4 Continuous features

Remaining features are all continuous. We plotted the graphs and found that for almost all features, the distribution of values are different for normal and attack instances respectively. Here you can find figures for two continuous variables - LIT401 and FIT101 in figures 5 and 6. There, however, seems to be some difference in patterns that occur for different variables. Therefore one key takeway is that not all continuous features are influenced in a similar way by the attacks but nearly every feature behaves differently - to lesser or to greater extent. This information makes feature selection a requirement. During attacks the distributions were more skewed.

Figure 5: FIT101: Distribution is different for attack and normal instances

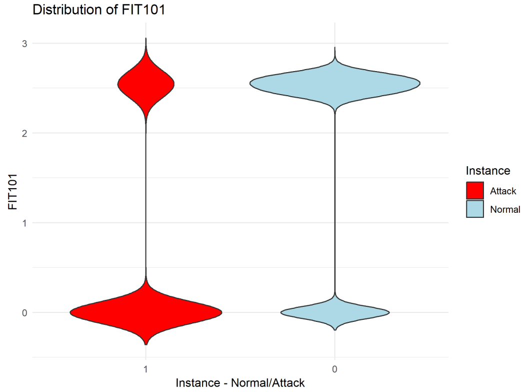
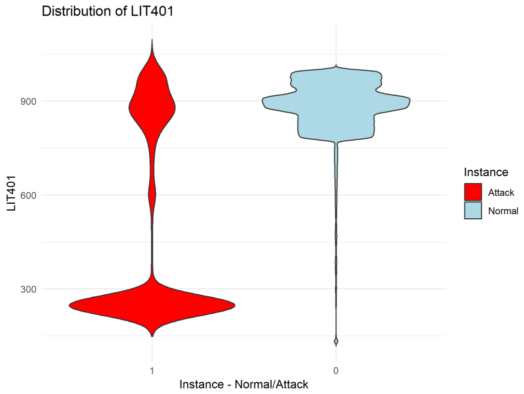


Figure 6: LIT401:Distribution is different for attack and normal instances



3.5 A look into some processes of the unit

As clear in figure 1, features MV101, T101 and P101 describe the process P1. MV101 is associated with raw water input, while LIT101 is the rating from the raw water tank and P101 is the measure from pump in P1. From figure 7, it is clear that initially there is a spike in the measure of LI101 as the tank is filling and this initiates the value of pump remains almost constant until the value of MV101 that is associated with the raw input increases. From figure 1, we can see that when AIT201 drops in value, we see a lot of fluctuations in the measurement from pump P201. This should be at-tributed to the fact that when the value of AIT201 drops, the value of chemical present in water drops and so P201 has to pump more HCL. The understanding of these processes is crucial during the feature selection so that we don't lose the features that are important. It is a cyber security prob-lem and therefore, without background understanding, the model may prove to be highly inaccurate. Process P3 and P4 were also explored but not detailed in this report.

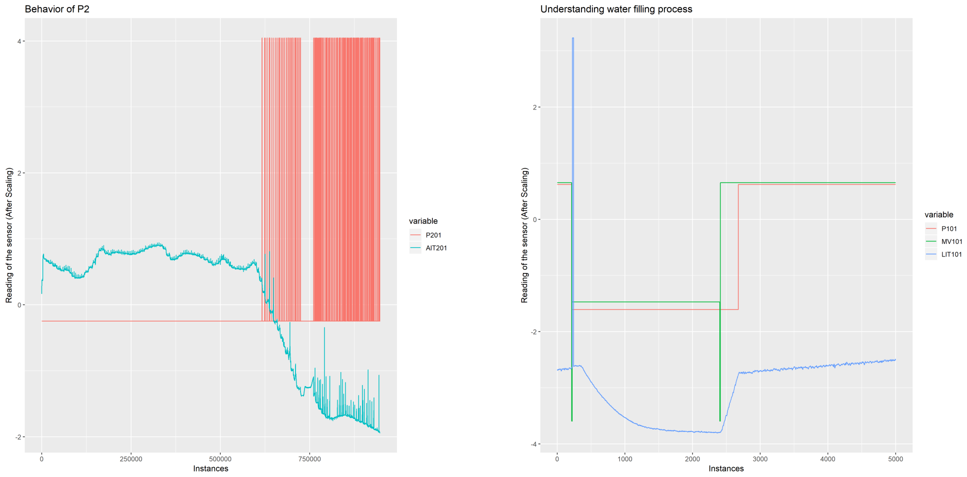


Figure 7: Chemical tank reading: AIT201 and pump P201(P2) (left), Water fillling process (P1) (right)

4 Sampling and Feature Selection

Figure 8: Process of Data Modelling

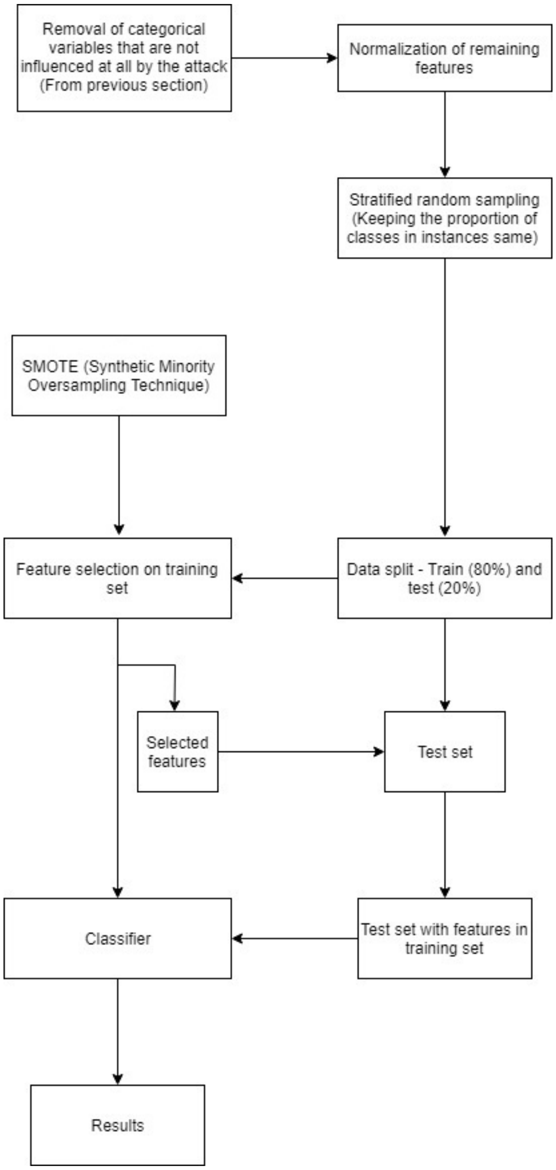

Removal of categorical variables that are not influenced at all by the attack (From previous section)
Normalization of remaining features
Stratified random sampling (Keeping the proportion of classes in instances same)
SMOTE (Synthetic Minority Oversampling Technique)
Feature selection on training set
Data split - Train (80%) and test (20%)
Selected features
Test set
Classifier
Test set with features in training set
Results

Figure 8 describes the process through feature selection and data modelling. Since measurements of our features are at different scales, it is necessary to scale them so as not to make one feature unnecessarily influence our results more than others. After normalization, we took a sample of about 50,000 instances from the dataset (due to limitation of our machine - Windows 8, 8 GB RAM). We used stratified random sampling to preserve the proportions of the normal and attack instances in the dataset. The sample dataset was then split randomly into two seperate datasets: training (80%) and testing (20%).

Since our dataset is imbalanced, we can do undersampling for the majority class (normal instances) and oversampling for the minority class (attack instances). One way to do it is Synthetic Minority Oversampling Technique (SMOTE) [4] from DMwR package of R [14]. To test the model accurately, the was done only on the training set. We saw earlier that not all features are highly influenced by attacks and therefore we employed a feature selection algorithm known as information.gain (Part of ID3) [1] from FSelector package. of R[12]. Figure 9 represents the features sorted by their importance score (Max. score =1). On investigating most important features as per the results of feature selection, we found that as exam-ined from previous section, there are many variables which have very less influence on the target variable. We also plotted correlation plot for continuous variables and while the correlation among some features was high, for other features it was too low to con-sider Principal Component Analysis (PCA) since PCA may result in under-influence of features that have very less correlation with other features and if those features are more important, it may give poor re-sults.

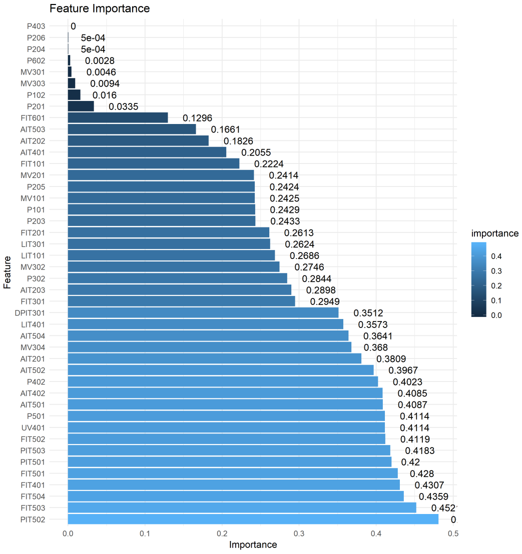
Therefore we chose to select features based on the observations from previous section and the scores of importance from the our information gain algorithm. These features are the features for which

$$IG_x > mean(IG)$$

where IG is information gain and x is a feature. Total of 26 features with high importance as seen in figure 9 were selected.

One important note is that the SMOTE and feature selection both were performed on the training sets, and therefore this does not influence the testing set at all and thus, making the evaluation from testing set valid.

Figure 9: Ranking of features as per importance



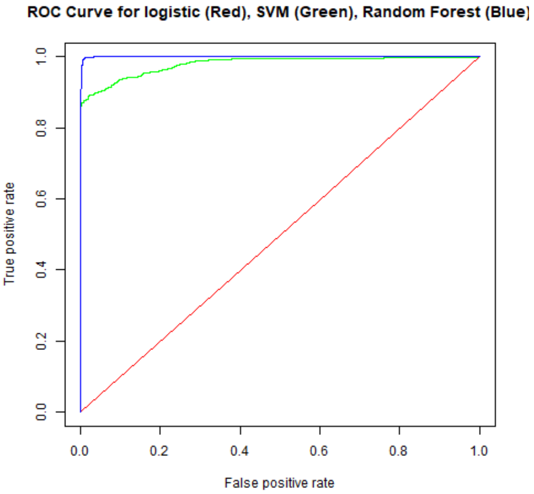
5 Model selection, evaluation and deployment

We tested our dataset on the test split of the sampled dataset that we processed in previous section. We used Multiple logistic regression, Support Vector Machine and Random Forest algorithm to model our training set. Since the dataset was labelled, the choice of supervised algorithm was made.

These models were selected as candidates because of following reasons:

1. Logistic Regression was used as a base model to see if we find improvement over it when we move to more complex models.
2. In cyber security applications, expecially when the dataset remains highly imbalanced, such as intruder detection, and detection of DDoS attacks, Random Forest, Support Vector Machine and neural network are good models as they tend to be more robust against the continuous fluctuations and imbalance or the data.

Figure 10: ROC: Logistic Regession (Red), SVM (Green), Random Forest (Blue)



The models for three different classifiers were trained on the training set that went through repeated undersampling and oversampling process. The training set included features selected from the feature selection process.

Random Forest was trained with 50 trees and the radial kernel was used to train SVM with tuned values for gamma and c. For gamma = 0.0625 and c = 4, the model performed best. For Random Forest, we utilized package randomForest [10] of R and for SVM, package e1071 [11] was utilized.

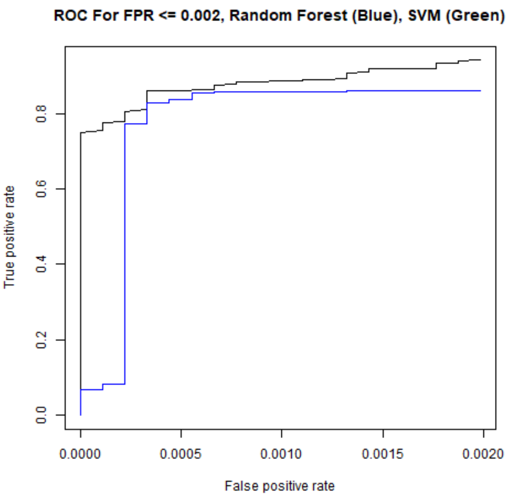
.

We tested the models on the test set, and the evaluation results are shown as Receiver Operating Characteristic (plot between True Positive Rate (TPR) and False Positive Rate (FPR)). Our primary metric is to have the best TPR (positives ∗ (1-specificity)) for a very low FPR (1-specificity)) in figure 10. For our Random Forest, the accuracy was about 99.5% and for our SVM, the accuracy was about 94%. Both these models performed far better than the logistic classifier.

It was clear that Logistic regression performed poorly in classifying instances accurately. We explored the results of SVM and Random Forest further. Since, this is a cyber security task and all the instances represent a single second of the day, it is essential to have a very low false positive rate so as not to cause false alarms. We set a threshold of 0.002 (False positive rate). We shall be interested in finding if the model is testing a high number of true positives for a false positive rate that causes false alarms for less than a minute per day, which accounts for around 2 seconds of false alarm every day. Figure 12 shows ROC for a FPR *leq* 0.002. A false alarm of less than 1 minute a day means that we should have FPR less than 0.0005 or 0.05%. For 0.05 % fpr, both SVM and Random Forest performed at similar levels. We chose Random Forest as it was consistently a good performer.
On R, we defined a function to return the optimal specificity and sensitivity for our model.

```
> print(find_optimal(perf, pred))
                 [,1]
sensitivity 0.9945770
specificity 0.9920687
cutoff      0.418000
```

Figure 11: ROC for FPR threshold (FPR *leq* 0.002)



For a false positive rate under 0.05 %, the model had a TPR of almost 83 %. This seems to be a reasonably well accurate model for our task. The false alarms (FPR) are reasonable for our task and attack detection (TPR) is reasonably high. On R, the model was saved for re-use.

6 Computation time

The entire computation took about 31.37 minutes. Most of the time were consumed during reading the dataset as the excel data files were large. This time includes making and saving visualizations as well.

```
> getduration <- endtime - starttime
> getduration
Time difference of 31.37299 mins
```

7 Conclusion

In conclusion, it can be said that there are significant relationships between chemical processes of water treatment unit and associated sensors. Further, values of various variables were affected by the attacks differently when compared to their values during the normal behavior of the unit. The correlations between features belonging to the same process stage was high while correlation among features belonging to different processes was quite low. So, after understanding these observations, we used a thorough methodology to select features that were important for our task. We used different classifiers for our task and since labels were available, the task was supervised. In the end, we were successful in our goal of identifying attacks.

References

[1] B Azhagusundari and Antony Selvadoss Thanamani. "Feature selection based on information gain". In: *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 2.2 (2013), pp. 18–21.

[2] Stuart A Boyer. *SCADA: supervisory control and data acquisition.* International Society of Automation, 2009.

[3] Richard J. Brennan. *Cyber attack on small Illinois water treatment plant has serious implications: security expert.* Nov. 2011. URL: `https://www.thestar.com/news/world/2011/11/21/cyber_attack_on_small_illinois_water_treatment_plant_has_serious_implications_security_expert.html/`.

[4] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[5] Jonathan Goh et al. "A dataset to support research in the design of secure water treatment systems". In: *International Conference on Critical Information Infrastructures Security.* Springer. 2016, pp. 88–99.

[6] Jonathan Goh et al. "Anomaly detection in cyber physical systems using recurrent neural networks". In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE).* IEEE. 2017, pp. 140–145.

[7] Jun Inoue et al. "Anomaly detection for a water treatment system using unsupervised machine learning". In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW).* IEEE. 2017, pp. 1058–1065.

[8] Khurum Nazir Junejo and Jonathan Goh. "Behaviour-based attack detection and classification in cyber physical systems using machine learning". In: *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security.* ACM. 2016, pp. 34–43.

[9] Moshe Kravchik and Asaf Shabtai. "Detecting cyber attacks in industrial control systems using convolutional neural networks". In: *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy.* ACM. 2018, pp. 72–83.

[10] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[11] David Meyer et al. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien.* R package version 1.7-0.1. 2019. URL: `https://CRAN.R-project.org/package=e1071`.

[12] Piotr Romanski and Lars Kotthoff. *FSelector: Selecting Attributes.* R package version 0.31. 2018. URL: `https://CRAN.R-project.org/package=FSelector`.

[13] Kerry Tomlinson. *Hackers change chemical settings at a water treatment plant.* Mar. 2016. URL: `https://archerint.com/how-hackers-changed-chemical-levels-in-peoples-drinking-water/`.

[14] L. Torgo. *Data Mining with R, learning with case studies.* Chapman and Hall/CRC, 2010. URL: `http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR`.