Detecting Attacks on a Water Treatment System Using Oneclass Support Vector Machines

Ken Yau, Kam-Pui Chow, Siu-Ming Yiu

Chapter 6

DETECTING ATTACKS ON A WATER TREATMENT SYSTEM USING ONE-CLASS SUPPORT VECTOR MACHINES

Ken Yau, Kam-Pui Chow and Siu-Ming Yiu

Abstract Critical infrastructure assets such as power grids and water treatment plants are monitored and managed by industrial control systems. Attacks that leverage industrial control systems to disrupt or damage infrastructure assets can impact human lives, the economy and the environment. Several attack detection methods have been proposed, but they are often difficult to implement and their accuracy is often low. Additionally, these methods do not consider the digital forensic aspects.

This chapter focuses on the use of machine learning, specifically one-class support vector machines, for attack detection and forensic investigations. The methodology is evaluated using a water treatment testbed, a scaled-down version of a real-world industrial water treatment plant. Data collected under normal operations and attacks are used in the study. In order to enhance detection accuracy, the water treatment process is divided into sub-processes for individual one-class support vector machine model training. The experimental results demonstrate that the trained sub-process models yield better detection performance than the trained complete process model. Additionally, the approach enhances the efficiency and effectiveness of forensic investigations.

Keywords: Machine learning, one-class SVM, forensics, water treatment system

1. Introduction

Industrial control systems, which combine distributed computing and physical process monitoring and control [9], are commonly used to operate critical infrastructure assets such as power grids and water treatment plants. Industrial control systems make it convenient to operate infrastructure assets remotely, but the added convenience comes at the cost of increased vulnerabilities [13]. Specifically, an attacker can compro-

mise a corporate network using conventional network security attacks and leverage the access to pivot and target industrial control systems. A widely-reported attack on a Ukrainian power grid in December 2015 caused a power outage to more than 200,000 customers [10]. The attackers leveraged spear phishing email, variants of the BlackEnergy 3 malware and Microsoft Office documents containing malware to penetrate information technology networks and launch attacks on electrical substations.

Digital forensics is increasingly engaging artificial intelligence to analyze large amounts of complex data [11]. Meanwhile, machine learning techniques have been shown to be very effective at detecting anomalies and attacks in industrial control systems. Supervised learning has yielded results with high precision, but the approach requires labeled (normal and attack) data for training. Class labeling is a challenging task because it is time consuming for large datasets and often requires manual efforts of the part of control system experts. Moreover, it is difficult or impossible to collect attack data. While some attacks may be simulated, it is not possible to simulate all possible attacks [19].

To address these challenges, this research employs a semi-supervised machine learning methodology in which a one-class support vector machine (OC-SVM) model is trained using normal data, following which data that deviate from the trained model are identified as attacks. This methodology does not need class labeling. Moreover, normal data for training is readily obtained.

An important aspect of the proposed methodology is that the physical process is divided into sub-processes and a one-class support vector machine model is created for each sub-process, which improves attack detection performance. Additionally, the division renders forensic investigations more effective. Instead of investigating the entire system at one time, a forensic practitioner can focus on individual sub-processes as needed. Since each trained sub-process model is responsible for detecting specific attacks, the practitioner is able to narrow the scope to perform data collection and investigate each sub-process individually. Experiments with a water treatment testbed demonstrate the improvements in attack detection and effectiveness of incident investigations.

2. Related Work

Attack detection in industrial control systems has been the subject of considerable research. Machine learning is one of the successful approaches for implementing attack detection.

Yau et al. [21, 22] have proposed forensic solutions for a simulated traffic light system that leverage machine learning techniques. They captured the values of relevant memory addresses used by the programmable logic controller that monitored and managed the traffic light system. The memory values were stored in a log file for model training and the trained model was used to identify anomalous programmable logic controller behavior. Although the solutions achieved high attack detection accuracy, the simulated system used in the research did not approach the scale and complexity of a real-world traffic light system.

Inoue et al. [8] have evaluated the application of unsupervised machine learning methods to anomaly detection in cyber-physical systems. Specifically, they compared two methods, deep neural networks and one-class support vector machines, for detecting anomalies in the same water treatment testbed used in this research. The results reveal that the two methods have various advantages and disadvantages with regard to detection performance and accuracy.

Mounce et al. [12] have employed supervised machine learning with support vector regression to detect novel events in time series data pertaining to water flow and pressure. The novel events include pipe bursts, hydrant flushing and sensor failure. Their research demonstrates that the methodology provides faster alert generation than approaches using artificial neural networks and fuzzy inference.

Schuster et al. [15] have applied one-class support vector machines to a number of real-world industrial control system traffic traces. Their experimental results show that one-class support vector machines are effective at analyzing network packets and packet sequences to detect anomalies.

Kravchik and Shabtai [9] have developed a methodology for detecting anomalies and attacks in industrial control systems using a 1D convolutional neural network and autoencoders. Convolutional neural networks are a popular machine learning technique used in image processing applications. An autoencoder is a neural network that is trained to reproduce its input, thereby learning useful properties of the data. Applications of the methodology to several popular public datasets reveal that the detection results match or exceed previously-published results while featuring a small footprint and short training and detection times, and providing more generality.

The methodology presented in this chapter is distinct from other approaches in that it divides a complex process into sub-processes for one-class support vector machine model training in order to increase attack detection performance and accuracy. The data used in this research was collected from a testbed that closely mimics a real-world water treat-

Figure 1. Secure Water Treatment (SWaT) testbed [3].

ment plant. A sliding window method is employed to process time series datasets for one-class support vector machine model training. Additionally, the methodology enhances forensic investigations of industrial control system incidents.

3. Secure Water Treatment Testbed

The Secure Water Treatment (SWaT) testbed shown in Figure 1 is set up at the iTrust Centre for Research in Cyber Security at Singapore University of Technology and Design. The testbed closely mimics a real-world water treatment plant [3]. The testbed takes raw water as input, executes a series of treatments and outputs recycled water.

The water treatment process comprises six sub-processes or stages P1 through P6 (Figure 2) [20]. Raw water enters the raw water tank (P1) from where it is pumped to chemical tanks. After chemical dosing and static mixing (P2), the water is passed to an ultrafiltration (UF) system (P3) and ultraviolet (UV) lamps (P4). Following this, the water is fed to a reverse osmosis (RO) system (P5). Finally, a backwash process cleans

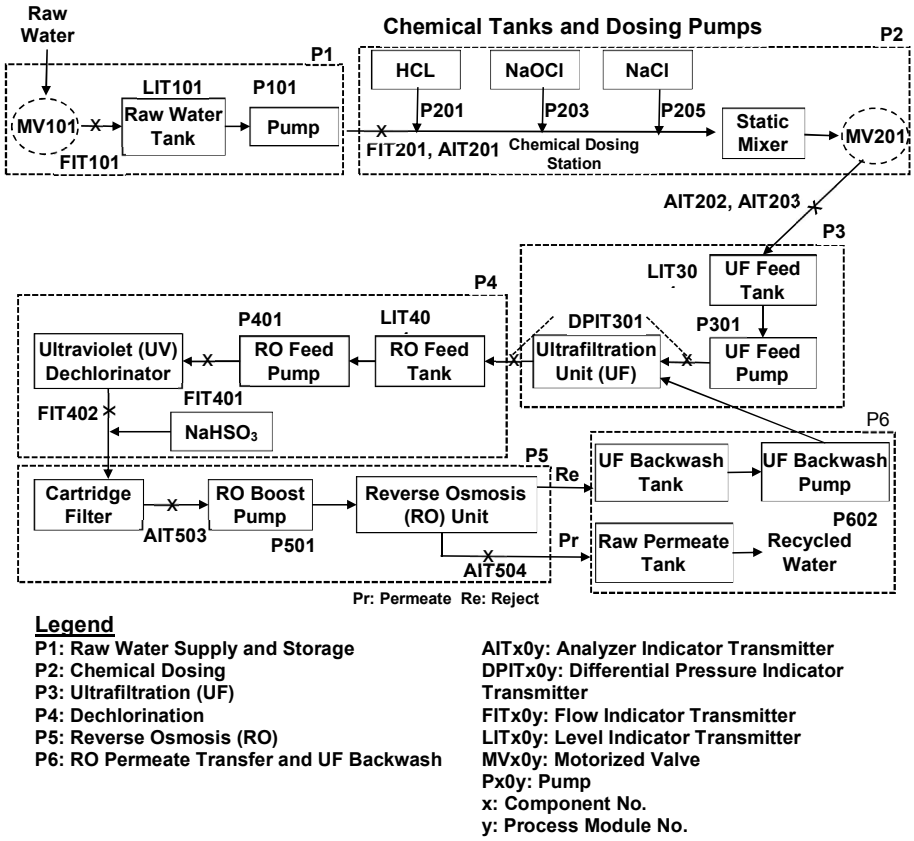


Figure 2. Six-stage water treatment process [3].

the membranes of the ultrafiltration system using the water produced by the reverse osmosis system (P6).

Sensors are employed at each sub-process; the sensor values are passed to a programmable logic controller, which monitors the states of the sub-processes. Based on the sensor values, the programmable logic controller directs actuators to manipulate the states of the sub-processes. For example, in the case of sub-process P1, the sensor LIT101 monitors the water level in the raw water tank. The programmable logic controller reads the sensor value and decides whether or not to change the state of the actuator, valve MV-101. If the LIT-101 sensor value is above a threshold, the programmable logic controller may deactivate valve MV-101, which stops raw water flow into the tank.

4. Data Collection

The data collection process lasted 11 days. The testbed was operated continuously 24 hours/day during the entire period. During the first seven days, the testbed operated under normal conditions (i.e., without attacks).

Attacks were launched during the last four days of the data collection process [7]. The attacks were created systematically from an attack model [1] that considers attacker intent. A total of 36 distinct attacks were launched on the SWaT testbed. The attacks fell in the following four categories [7]:

- **Single Stage Single Point (SSSP):** This type of attack targets one point in a single stage (sub-process).
- **Single Stage Multi Point (SSMP):** This type of attack targets two or more attack points in a single stage (sub-process).
- **Multi Stage Single Point (MSSP):** This type of attack targets one point in multiple stages (sub-processes).
- **Multi Stage Multi Point (MSMP):** This type of attack targets two or more points in multiple stages (sub-processes).

Data from all the testbed sensors and actuators was logged every second and stored in a historian. A total of 946,722 data samples involving 51 attributes (e.g., FIT101, LIT101 and P101) were collected over the 11-day period. Figure 3 shows sample data that was collected during the experiments.

5. One-Class Support Vector Machine

Machine learning builds an automated analytical model using algorithms that learn from data iteratively. Based on the model, machine learning enables the automated discovery of hidden insights without explicit programming [14]. A one-class support vector machine is a semi-supervised learning model that is widely used to detect anomalous events. The one-class support vector machine essentially finds the maximal margin hyperplane using an appropriate kernel function to map most of the training data to one side of the hyperplane [2]. Thus, it is trained using only data from only one (normal) class. After being trained with normal data, the one-class support vector machine classifies test data as normal data or abnormal (i.e., attack) data.

Timestamp	FIT101	LIT101	MV101	P101	P102	AIT201	AIT202	AIT203	FIT201	...
22/12/2015 4:30:00 PM	0	124.3135	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:01 PM	0	124.392	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:02 PM	0	124.4705	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:03 PM	0	124.6668	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:04 PM	0	124.5098	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:05 PM	0	123.921	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:06 PM	0	123.5284	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:07 PM	0	123.4107	1	1	1	251.9226	8.313446	312.7916	0	...
22/12/2015 4:30:08 PM	0	123.2144	1	1	1	251.9226	8.312805	312.7916	0	...
22/12/2015 4:30:09 PM	0	123.3322	1	1	1	251.9226	8.310242	312.7916	0	...
22/12/2015 4:30:10 PM	0	123.7247	1	1	1	251.9226	8.30896	312.8685	0	...
22/12/2015 4:30:11 PM	0	124.2742	1	1	1	251.9226	8.30896	312.9198	0	...
22/12/2015 4:30:12 PM	0	124.4705	1	1	1	251.9226	8.30896	312.9198	0	...
22/12/2015 4:30:13 PM	0	124.863	1	1	1	251.9226	8.30896	312.9198	0	...
22/12/2015 4:30:14 PM	0	125.0593	1	1	1	251.9226	8.30896	312.9198	0	...
22/12/2015 4:30:15 PM	0	124.5883	1	1	1	251.9226	8.30896	312.9198	0	...
22/12/2015 4:30:16 PM	0	124.392	1	1	1	251.9226	8.30896	312.9198	0	...

Figure 3. Sample data.

6. Methodology

In the experiments, data from the first seven days (without attacks) was used to train the one-class support vector machine. Data from the last four days (with attacks) was used to evaluate the one-class support vector machine performance.

Since the scales of the various testbed features (attributes) were different (Figure 3), the min-max scaling method was used to normalize the values of the features to a scale of 0 to 1 in order to achieve better model training performance. Min-max scaling is performed as follows:

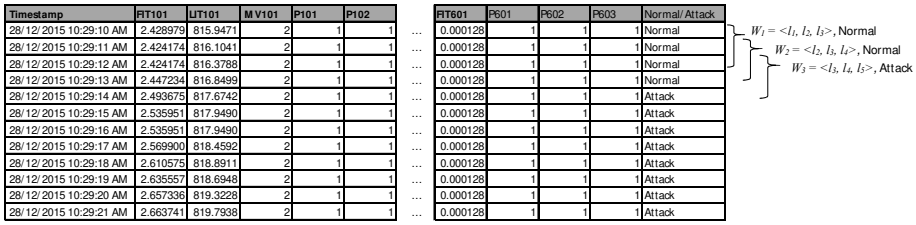
$$x^{'} = \frac{x - min(x)}{max(x) - min(x)}$$

where x is the original value and $x^{'}$ is the normalized value.

Since the data was logged as a time series, the sliding window method was used to convert the data into individual feature vectors [6, 8]. Assume that l_i is the i^{th} log entry and w is the window size, then the window W_i is given by:

$$W_i = l_i, l_{i+1}, \ \ldots, \ l_{i+w-1}$$

If there are k entries $l_1, l_2, \ldots, l_k$, then $k - w + 1$ windows $W_i, W_2, \ldots, W_{k-w+1}$ are generated. A window is labeled as an attack window if at



Timestamp	FIT101	LIT101	MV101	P101	P102	...	FIT601	P601	P602	P603	Normal/Attack
28/12/2015 10:29:10 AM	2.428979	815.9471	2	1	1	...	0.000128	1	1	1	Normal
28/12/2015 10:29:11 AM	2.424174	816.1041	2	1	1	...	0.000128	1	1	1	Normal
28/12/2015 10:29:12 AM	2.424174	816.3788	2	1	1	...	0.000128	1	1	1	Normal
28/12/2015 10:29:13 AM	2.447234	816.8499	2	1	1	...	0.000128	1	1	1	Normal
28/12/2015 10:29:14 AM	2.493675	817.6742	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:15 AM	2.535951	817.9490	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:16 AM	2.535951	817.9490	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:17 AM	2.569900	818.4592	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:18 AM	2.610575	818.8911	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:19 AM	2.635557	818.6948	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:20 AM	2.657336	819.3228	2	1	1	...	0.000128	1	1	1	Attack
28/12/2015 10:29:21 AM	2.663741	819.7938	2	1	1	...	0.000128	1	1	1	Attack

Figure 4. Sliding log entries into windows of size three.

least one of the log entries $l_i, l_{i+1}, \ldots, l_{i+w-1}$ in the window is labeled as an attack; otherwise, the window is labeled as a normal window.

Figure 4 shows how the log entries slide into windows of size three. Each window is fed to the trained model to classify it as normal or attack. The experiments compared the trained model performance achieved for different window sizes.

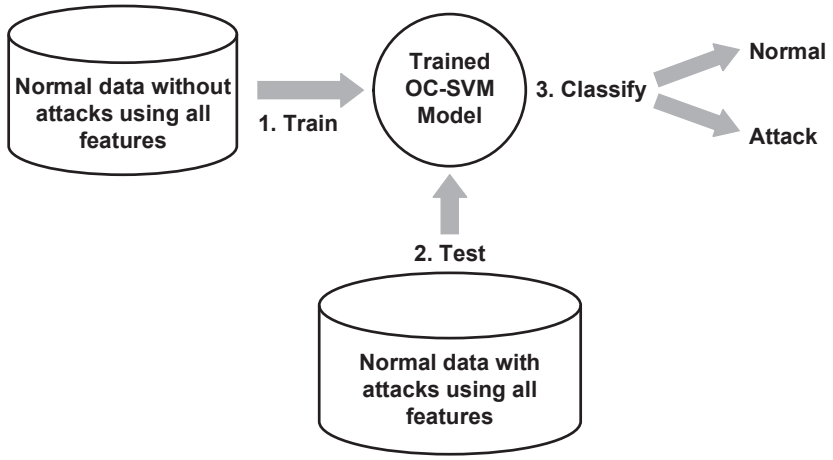


Figure 5. Approach 1: Model training for the entire SWaT process.

In general, there are two approaches for model training. Approach 1 creates a trained a model using the entire process with all the data features (Figure 5). The trained model is then used to determine if any attacks were launched against the water treatment system. However, this approach cannot identify the sub-processes that were attacked.

The second approach, Approach 2, trains the models for the sub-processes separately using their own features. Figure 6 shows the details of the approach. For example, the model M_{P1} is trained using only

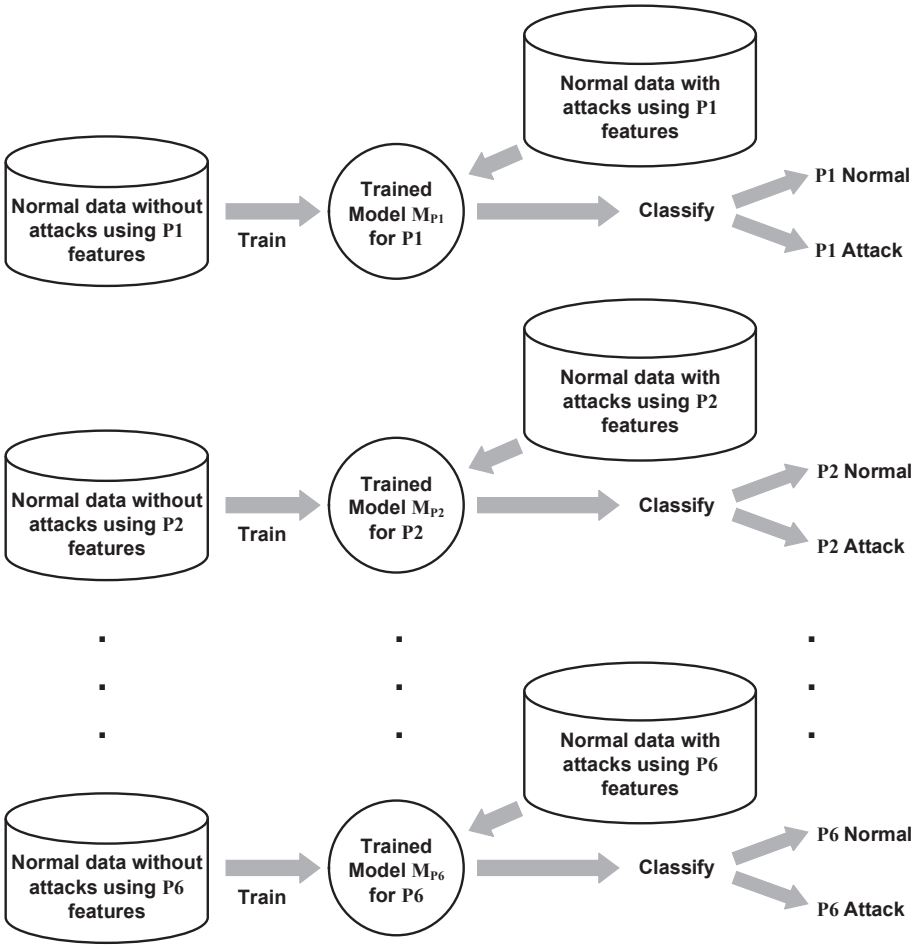


Figure 6. Approach 2: Model training for each of the six SWaT sub-processes.

P1 features (FIT101, LIT101, MV101, P101 and P102). The trained model M_{P1} is then used to detect attacks on sub-process P1. The one-class support vector machine used in the experiments was implemented using the scikit-learn machine learning library [16] on TensorFlow [18], an end-to-end open source machine learning platform.

Optimum one-class support vector machine classifiers for attack detection were realized using the parameters: (i) nu = 10^{-5}; (ii) gamma = auto; and (iii) kernel = sigmoid. Note that nu is an upper bound on the fraction of training errors and a lower bound on the fraction of support vectors; gamma defines the influence of a single training sample

(default value is auto, which corresponds to the reciprocal of the number of features); and the kernel type may be linear, poly, rbf or sigmoid.

7. Evaluation and Experimental Results

This section describes the evaluation procedure and the experimental results.

7.1 Evaluation

Since an imbalance exists between normal and attack data in the testing dataset, it is not appropriate to measure the performance of a one-class support vector machine model using the accuracy metric (i.e., number of correct predictions from among all predictions made). In the case of imbalanced datasets, when the minority (attack) class is an important class, the performance metrics suggested by Bekkar et al. [4] are more appropriate. These metrics are based on a confusion matrix that reports the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

Precision, recall and F-score were used to evaluate the performance of a classifier on the minority class [5, 17]:

- **Precision:** This measure is defined as the number of correctly classified positive samples divided by the number of samples labeled by the system as positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall:** This measure is defined as the number of correctly classified positive samples divided by the number of all relevant samples (i.e., all the samples that should have been identified as positive):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F-score:** This measure is defined as the harmonic mean of the precision and recall:

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

7.2 Experimental Results

According to Dietterich [6], the sliding window method converts a sequential supervised learning problem into a classical supervised learning problem, which yields adequate performance in many applications.

Table 1. Approach 1 classification performance.

Window Size	Precision (%)	Recall (%)	F-score (%)
N/A	88.07	99.96	93.63
3	88.08	99.96	93.64
5	88.08	99.96	93.64

Table 2. Approach 2 classification performance.

Sub-Process	Window Size	Precision (%)	Recall (%)	F-score (%)
P1 (5 Features)	N/A	98.28	99.86	99.07
	3	98.28	99.86	99.06
	5	98.28	99.86	99.06
P2 (11 Features)	N/A	99.25	100.00	99.63
	3	99.25	100.00	99.63
	5	99.25	100.00	99.63
P3 (9 Features)	N/A	90.29	100.00	94.89
	3	90.44	99.98	94.97
	5	90.47	99.95	94.98
P4 (9 Features)	N/A	99.07	99.89	99.48
	3	99.07	99.89	99.48
	5	99.08	99.89	99.48
P5 (13 Features)	N/A	99.49	99.91	99.70
	3	99.49	99.91	99.70
	5	99.49	99.91	99.70
P6 (4 Features)	N/A	99.82	99.99	99.91
	3	99.82	99.99	99.91
	5	99.82	99.99	99.91

However, the experimental results demonstrate that the sliding window method applied to the entire process (Approach 1 in Table 1) and to individual sub-processes (Approach 2 in Table 2) does not improve the performance of the one-class support vector machine classifiers for attack detection when the window sizes are set to three and five. On the other hand, the precision and F-score values are significantly increased for Approach 2 (Table 2), which divides the entire process into six sub-processes for model training. Note that identical parameter settings (nu $= 10^{-5}$, gamma = auto and kernel = sigmoid) were employed for one-class support vector machine training in Approach 1 and Approach 2.

An advantage of the proposed methodology is that the parameter settings can be adjusted individually for training each sub-process classifier in order to achieve the best performance. Moreover, this methodology re-

sults in better attack detection performance, and increases the efficiency and effectiveness of forensic investigations. Since the timestamps, durations and activity sequences of sub-process attacks are recorded in a log file during the classification process (Figure 6), a forensic investigator is able to obtain more evidence about the case from the classification log file when the attack log activities and time sequences are correlated with other system/network logs and the activity logs on a suspect's computer.

8. Conclusions

Detecting and investigating attacks on industrial control systems are vital to securing critical infrastructure assets. This chapter has described a semi-supervised machine learning methodology in which a one-class support vector machine model is trained using normal data, following which attacks are identified as data that deviates from the trained model. The methodology eliminates the need to employ labeled (normal and attack) data for training – class labeling is time consuming for large datasets and it is difficult, if not impossible, to collect attack data. Another important aspect is that the methodology divides a physical process into sub-processes, and a one-class support vector machine model is created for each sub-process.

Experimental results using a water treatment testbed demonstrate that the trained sub-process models yield better attack detection performance than the trained complete process model. Additionally, the division into sub-processes renders forensic investigations more effective. Instead of investigating the entire system at one time, a forensic practitioner can focus on individual sub-processes as needed. Since each trained sub-process model is responsible for detecting specific attacks, the practitioner is able to narrow the scope to perform data collection and investigate each sub-process individually.

Future research will attempt to improve attack detection performance using machine learning on large, real-world datasets. Additionally, it is will attempt to use artificial intelligence techniques to support forensic investigations of industrial control systems.

Acknowledgement

The authors wish to thank the iTrust Centre for Research in Cyber Security at Singapore University of Technology and Design for providing the datasets used in this research.

References

[1] S. Adepu and A. Mathur, An investigation into the response of a water treatment system to cyber attacks, *Proceedings of the Seventeenth IEEE International Symposium on High Assurance Systems Engineering*, pp. 141–148, 2016.

[2] S. Amraee, A. Vafaei, K. Jamshidi and P. Adibi, Abnormal event detection in crowded scenes using a one-class SVM, *Signal, Image and Video Processing*, vol. 12(6), pp. 1115–1123, 2018.

[3] K. Aung, Secure Water Treatment Testbed (SWaT): An Overview, iTrust Centre for Research in Cyber Security, Singapore University of Technology and Design, Singapore, 2015.

[4] M. Bekkar, K. Djemaa and T. Alitouche, Evaluation measures for model assessment over imbalanced datasets, *Journal of Information Engineering and Applications*, vol. 3(10), pp. 27–38, 2013.

[5] A. Bottenberg and J. Ward, Applied Multiple Linear Regression, Technical Documentary Report PRL-TDR-63-6, Air Force Systems Command, Lackland Air Force Base, Texas, 1963.

[6] G. Dietterich, Machine learning for sequential data: A review, *Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition, and Structural and Syntactic Pattern Recognition*, pp. 15–30, 2002.

[7] J. Goh, S. Adepu, K. Junejo and A. Mathur, A dataset to support research in the design of secure water treatment systems, *Proceedings of the International Conference on Critical Information Infrastructures Security*, pp. 88–99, 2016.

[8] J. Inoue, Y. Yamagata, Y. Chen, M. Poskitt and J. Sun, Anomaly detection in a water treatment system using unsupervised machine learning, *Proceedings of the IEEE International Conference on Data Mining Workshops*, pp. 1058–1065, 2017.

[9] M. Kravchik and A. Shabtai, Efficient Cyber Attack Detection in Industrial Control Systems using Lightweight Neural Networks, Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel, 2019.

[10] M. Lee, M. Assante and T. Conway, Analysis of the Cyber Attack on the Ukrainian Power Grid, TLP: White, SANS Industrial Control Systems, Bethesda, Maryland, and Electricity Information Sharing and Analysis Center, Washington, DC, 2016.

[11] F. Mitchell, The use of artificial intelligence in digital forensics: An introduction, *Digital Evidence and Electronic Signature Law Review*, vol. 7, pp. 35–41, 2010.

[12] S. Mounce, R. Mounce and J. Boxall, Novelty detection for time series data analysis in water distribution systems using support vector machines, *Journal of Hydroinformatics*, vol. 13(4), pp. 672–686, 2011.

[13] D. Ramotsoela, A. Abu-Mahfouz and G. Hancke, A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study, *Sensors*, vol. 18(8), article E2491, 2018.

[14] SAS Institute, Machine learning: What it is and why it matters, Cary, North Carolina (`www.sas.com/en_us/insights/analytics/machine-learning.html`), 2019.

[15] F. Schuster, A. Paul, R. Rietz and H. Koenig, Potential of using a one-class SVM for detecting protocol-specific anomalies in industrial networks, *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 83–90, 2015.

[16] scikit-learn, Machine learning in Python (`scikit-learn.org`), 2019.

[17] M. Sokolova and G. Lapalme, A systematic analysis of performance measures for classification tasks, *Information Processing and Management*, vol. 45(4), pp. 427–437, 2009.

[18] TensorFlow, TensorFlow: An end-to-end open source machine learning platform (`www.tensorflow.org`), 2019.

[19] R. Vlasveld, Introduction to One-Class Support Vector Machines (`rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines`), July 12, 2013.

[20] J. Wang, J. Sun, Y. Jia, S. Qin and Z. Xu, Towards "verifying" a water treatment system, in *Formal Methods*, K. Havelund, J. Peleska, B. Roscoe and E. de Vink (Eds.), Springer, Cham, Switzerland, pp. 73–92, 2018.

[21] K. Yau and K. Chow, PLC forensics based on control program logic change detection, *Journal of Digital Forensics, Security and Law*, vol. 10(4), pp. 59–68, 2015.

[22] K. Yau and K. Chow, Detecting anomalous programmable logic controller events using machine learning, in *Advances in Digital Forensics XIII*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 81–94, 2017.