Security in Process:
Detecting Attacks in Industrial Process Data

Simon D. Duque Anton
simon.duque_anton@dfki.de
German Research Center for AI
Intelligent Networks Research Group
Kaiserslautern, Germany

Anna Pia Lohfink
lohfink@cs.uni-kl.de
University of Kaiserslautern
Department of Computer Science
Kaiserslautern, Germany

Christoph Garth
garth@cs.uni-kl.de
University of Kaiserslautern
Department of Computer Science
Kaiserslautern, Germany

Hans Dieter Schotten
hans_dieter.schotten@dfki.de
German Research Center for AI
Intelligent Networks Research Group
Kaiserslautern, Germany

ABSTRACT

Due to the fourth industrial revolution, industrial applications make use of the progress in communication and embedded devices. This allows industrial users to increase efficiency and manageability while reducing cost and effort. Furthermore, the fourth industrial revolution, creating the so-called *Industry 4.0*, opens a variety of novel use and business cases in the industrial environment. However, this progress comes at the cost of an enlarged attack surface of industrial companies. Operational networks that have previously been phyiscally separated from public networks are now connected in order to make use of new communication capabilites. This motivates the need for industrial intrusion detection solutions that are compatible to the long-term operation machines in industry as well as the heterogeneous and fast-changing networks. In this work, process data is analysed. The data is created and monitored on real-world hardware. After a set up phase, attacks are introduced into the systems that influence the process behaviour. A time series-based anomaly detection approach, the *Matrix Profiles*, are adapted to the specific needs and applied to the intrusion detection. The results indicate an applicability of these methods to detect attacks in the process behaviour. Furthermore, they are easily integrated into existing process environments. Additionally, one-class classifiers *One-Class Support Vector Machines* and *Isolation Forest* are applied to the data without a notion of timing. While *Matrix Profiles* perform well in terms of creating and visualising results, the one-class classifiers perform poorly.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; *Network security*.

KEYWORDS

Anomaly Detection, Intrusion Detection, Industrial Networks, Machine Learning, Time Series

1 INTRODUCTION

Attacks on industrial enterprises have increased over the last two decades, in frequency as well as in impact [5]. A number of differences between home and office Information Technology (IT) networks makes classic IT security solutions only partly applicable to the industrial environment. Heterogeneous networks, consisting of physically distributed devices with operation times of several decades make updates and fixes difficult. Communication protocols such as *Modbus/TCP* [24, 25] are employed that do not contain means of authentication or encryption. That means any attacker having obtained access to the communication network is capable of reading and injecting messages. For two historic reasons, industrial, Supervisory Control And Data Acquisition (SCADA) or Operation Technology (OT) networks are not secured in a fashion deemed appropriate for home and office IT [16]. First, OT networks are supposed to be physically separated from IT networks. Second, OT networks and their connected devices are considered to be highly application specific, thus making propagation and exploitation by an attacker difficult. Both reasons do not hold true anymore. Commercial Off-The-Shelf (COTS) products in the industrial area such as Programmable Logic Controllers (PLCs) make set up, maintenance and operation of industrial applications much easier due to common interfaces and programming libraries. Those, however, make it easier for an attacker to prepare for exploitation as well. Furthermore, the fourth industrial revolution introduces an abundance of novel use and business cases [1]. Most of them rely on the communication and computation capabilities of Internet of Things (IoT) and Industrial Internet of Things (IIoT) devices. This breaks the physical separation of networks, creating access routes

to OT networks. Even if no such access is possible, attackers have successfully managed to move laterally to the OT networks in the past after breaking the IT network perimeter [7]. As a result, efficient intrusion detection in industrial networks is crucial for the secure and sound operation of industrial applications. However, several attacks are capable of masking sensor outputs once the attacker gains access to the device under attack. Thus, correlation of several sensor values, preferably over separate channels, can be employed to detect attacks.

In this work, machine learning- as well as time series-based methods for anomaly detection are applied to one set of process data of a real-world industrial process, containing several sensors and actuators, controlled by six PLCs. As the data was created and captured in a experimental environment, ground truth about the attacks is available, i.e. there are labels and the assurance that the data is labelled correctly. The contribution of this paper is twofold:

- The feasibility of multi-sensor anomaly detection based on machine learning is evaluated
- a time series motif discovery algorithm is extended for anomaly detection

The remainder of this work is structured as follows. In Section 2, an overview of the state of the art is provided. The data set analysed in this work are presented in Section 3, the algorithms used to evaluate the data are introduced in Section 4. In Section 5, they are evaluated. This work is concluded in Section 6.

2 RELATED WORK

Due to the increasing relevance of industrial intrusion detection, it is a widely regarded topic in the research community. *Schneider and Böttinger* use *autoencoders* in order to detect real attacks in an industrial data set in an unsupervised fashion [30]. A framework for assessing the impact of cyber attacks in production environments is presented by *Giehl et al.* [13]. They evaluate their approach on the *SUTD Security Showdown (S3) 2017* (*S317*) data set provided by the *iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design* [17]. The attack data has been generated by a contest to affect the process environment with real attacks. *Goh et al.* use Recurrent Neural Networks (RNNs) to detect attacks on a Cyber-Physical Systems (CPSs) in a data set by the same institution [15]. They evaluate the *Secure Water Treatment* (*SWaT*) data set [18], the same data set that has been evaluated in the course of this work. Creating realistic data sets for industrial intrusion detection is crucial for the development and testing of intrusion detection systems. A special kind of neural networks, *Long Short-Term Memory* (*LSTM*), is used by *Feng et al.* [9]. They employ a multi-level approach in order to detect attacks in a gas pipeline data set. *Knapp and Langill* present approaches to secure industrial networks [19]. The detection of intrusions in power system networks is discussed by *Yang et al.* [32]. *Larkin et al.* present a summary about the evolution of SCADA security systems [20]. *One-Class Support Vector Machine* (*OCSVM*) as a machine learning algorithm to detect novel and unknown attacks is presented by *Maglaras and Jiang* [23]. Approaches to detect attacks in *Modbus* data with the help of signatures is presented by *Gao and Morris* [12]. The security of future industrial applications with the integration of the IIoT is addressed by *Plaga et al.* [28, 29].
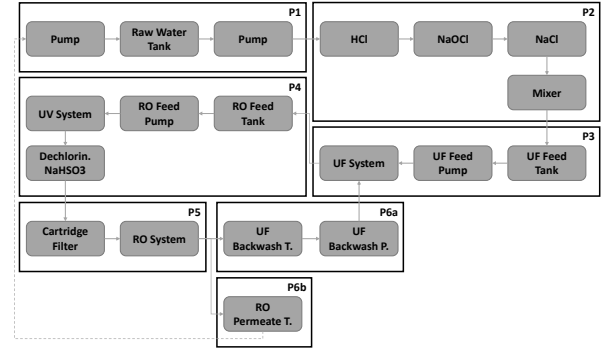

Figure 1: Relation of Sub-Processes

3 DATA SET

An industrial data set is evaluated in this work. It is presented by the *iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design* and is called *SWaT* [14, 18]. It is taken in an industrial environment with a real-world underlying process into which attacks have been introduced. The data set is gathered from a water processing facility that was set up in a laboratory context. It consists of the following six sub-processes:

- *P1*: Raw water storage
- *P2*: Pre-treatment
- *P3*: Membrane Ultra Filtration (UF)
- *P4*: Dechlorination by Ultraviolet (UV) lamps
- *P5*: Reverse Osmosis (RO)
- *P6*: Disposal

The relations of and transitions between the sub-processes are depicted in Figure 1. First, the water to be processed is stored, followed by a pre-treatment with different chemicals. After that, UF is applied, followed by an UV process. It is then pumped to an RO process. Depending on the level of cleanliness, it is either stored in a clean water reservoir or fed back to the UF process. The environment in which the PLCs controlling the sub-processes are set up is depicted in Figure 2. The PLCs in this figure correspond to the sub-processes with the respective number. In total, they control 51 sensors and actuators. An exhaustive list including description can be found in the work of *Goh et al.* [14]. They distinguish between four types of attacks, describing the type and listing the number of occurrences in the data:

- *Single Stage Single Point (SSSP)*: Single stage attack on one point in the process, 26 instances in the data set
- *Single Stage Multi Point (SSMP)*: Single stage attack on multiple points in the process, 4 instances in the data set
- *Multi Stage Single Point (MSSP)*: Multi stage attack on one point in the process, 2 instances in the data set
- *Multi Stage Multi Point (MSMP)*: Multi stage attack on multiple points in the process, 4 instances in the data set

4 ALGORITHMS USED

In this work, three different algorithms have been used to determine attacks in the data set: *OCSVM*s, *Isolation Forests* and *Matrix Profiles*. *OCSVM*s and *Isolation Forests* have been used to analyse the data
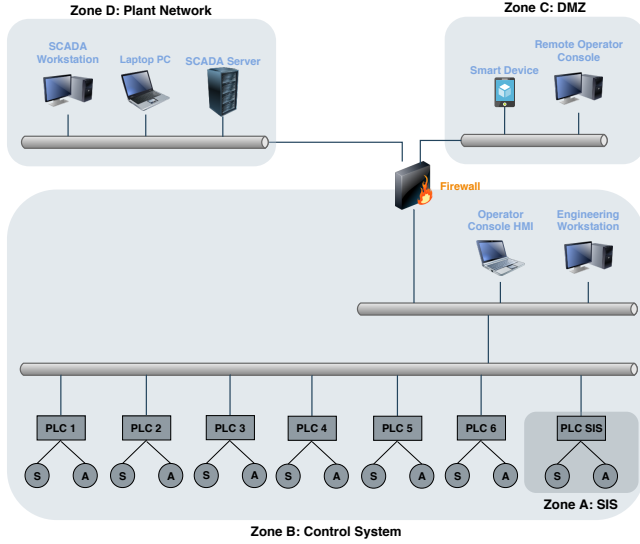
Figure 2: Schematic Overview of the Process Environment

sets, as described in Section 3, on packet basis while *Matrix Profiles* have been used to perform a time series analysis of the data sets.

4.1   One-Class Support Vector Machines

*OCSVM*s are a one-class classifier that is trained with one class in order to determine whether elements of the test case do or do not belong to that class. It is well suited for applications where one class is predominant, such as anomaly detection. *Support Vector Machine*s (*SVM*s) are a large-margin classifier. They were introduced by *Boser et al.* in 1992 [3]. Elements in an n-dimensional data space are separated by an $n + 1$-dimensional hyperplane. The elements are described as tuples as shown in (1) [4].

$$(x_i, y_i), i = 1, ..., m, y \in \{-1, 1\} \tag{1}$$

x is a vector of dimension n, y is an attribution indicating affiliation to one of two classes. In classic *SVM*, both classes are present in the training data, leading to a hyperplane separating both classes so that the distance of every element to the hyperplane is maximal. In one-class classification according to *Schölkopf et al.* [31], the distances of data points from the hyperplane are maximised. A quadratic minimisation function (2) minimises the area surrounded by the hyperplane according to v.

$$\min_{\omega, \xi_i, \rho} \frac{1}{2} \|\omega\|^2 + \frac{1}{vn} \sum_{i=1}^{n} \xi_i - \rho$$

$$\text{with} \tag{2}$$

$$(\omega \times \phi(x_i)) \geq \rho - \xi_i \forall i \in \{1, ..., n\}$$

$$\xi_i \geq 0 \forall i \in \{1, ..., n\}$$

$v \in (0, 1)$ characterises the fraction of outliers that are tolerated as well as the training size. ω and ρ are used to describe the hyperplane. After training, the decision function (3) decides whether or not an element is part of the trained class or not.

$$f(x) = \text{sgn}((\omega \times \phi(x_i) - \rho) \tag{3}$$

4.2   Isolation Forest

Similar to *OCSVM*, *Isolation Forest* is a one-class classifier [21]. The goal is to isolate anomalous data points in a data set. This is done by training an ensemble of decision trees on a data set and then considering the path length until convergence as a metric for isolation. Densely populated areas converge quickly, while isolated areas take longer to converge. A data sample is described by (4).

$$X = \{x_1, ..., x_n\} \tag{4}$$

A tree is created by selecting random attributes q as well as a split value p until either:

- the height limit of the tree is reached,
- $|X| = 1$ or
- all data in the tree is of the same value.

The average path length is calculated and derived from this, the path length of each sample is calculated and an anomaly score can be derived. In sorting the path lengths in ascending order, anomalies are found at the top of the list.

4.3   Matrix Profiles

Matrix Profiles were developed in 2016 by *Yeh et al.* [33] as an algorithm for motif discovery. A time series data set was split into sequences of length m. The distance of each sequence starting at a point in the data set from each other sequence is then calculated in a sliding window fashion, e.g. with the z-normalised distance (5).

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (\hat{x}_i - \hat{y}_i)^2}$$

$$\hat{x}_i = \frac{x_i - \mu_x}{\sigma_x}, \quad \hat{y}_i = \frac{y_i - \mu_y}{\sigma_y} \tag{5}$$

After applying *Pearson's Correlation Coefficient* [2] (6)

$$corr(x, y) = \frac{E((x - \mu_x)(y - \mu_y))}{\sigma_x \sigma_y}$$

$$= \frac{\sum_{i=1}^{m} x_i y_i - m\mu_x \mu_y}{m\sigma_x \sigma_y}, \tag{6}$$

where

$$\mu_x = \frac{\sum_{i=1}^{m} x_i}{m}, \quad \mu_y = \frac{\sum_{i=1}^{m} y_i}{m} \tag{7}$$

and

$$\sigma_x^2 = \frac{\sum_{i=1}^{m} x_i^2}{m} - \mu_x^2, \quad \sigma_y^2 = \frac{\sum_{i=1}^{m} y_i^2}{m} - \mu_y^2. \tag{8}$$

The Euclidean distance relates as in (9) [26],

$$d(x, y) = \sqrt{2m(1 - corr(x, y))} \tag{9}$$

the resulting metric for distance calculation is described in (10).

$$d(x, y) = \sqrt{2m\left(1 - \frac{\sum_{i=1}^{m} x_i y_i - m\mu_x \mu_y}{m\sigma_x \sigma_y}\right)} \tag{10}$$

x and y are time series, μ is the respective mean and σ the respective standard deviation. The minimal distances are derived and stored in a matrix, hence the name. A high minimal distance indicates an outlier, as no sequence in the time series is similar. Correspondingly,

a low minimal distance indicates at least one similar sequence in the series.

5 EVALUATION

In this section, the application of the algorithms introduced in Section 4 on the data set as presented in Section 3 is evaluated. All algorithms employed in the course of this work only need to be trained on normal data. This is based on the assumption that in real applications, anomalous data is sparse. Furthermore, since the anomalies of interest in this work are due to attacks, they might be unique in their characteristic and thus hard to train a priori. Having one-class classifiers or predictors is justified by the reality of industrial environments that data from normal operation of productive systems is available in abundance while anomalous data is hardly present.

5.1 *Matrix Profiles*

In this work, *Matrix Profiles* are applied to the data set in order to determine thresholds of the minimal distance as well as create an additional metric: the number of similar instances. Preliminary work shows the effectiveness of *Matrix Profiles* for the detection of attacks in process behaviour [8]. The reference implementation of *Matrix Profiles* was used and customised for the application purposes. In Figure 3, the level of raw water tank *LIT-301* is shown. In this figure, about 100 000 time steps of normal behaviour are
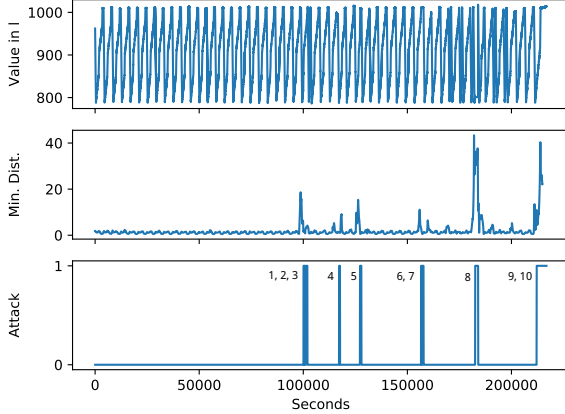

Figure 3: *Matrix Profile* of LIT-301

preceding about 150 000 time steps of behaviour during which ten attacks occur. The attacks are indicated and numbered in the bottom row by a boolean value 1. Additionally, the value of the sensor is shown in the first row and the minimal distance as calculated with the *Matrix Profile* is shown in the second row. It can be seen that all attacks can be detected by the increase in minimal distance as calculated with the *Matrix Profile*. The hyper-parameter m was set to 500. Additionally, a sensor measuring differential pressure in the backwash-process *DPIT-301* was analysed in the same fashion, shown in Figure 4. Its behaviour is more rugged than the behaviour of *LIT-301*, the hyper-parameter m is set to 2000, the period of the
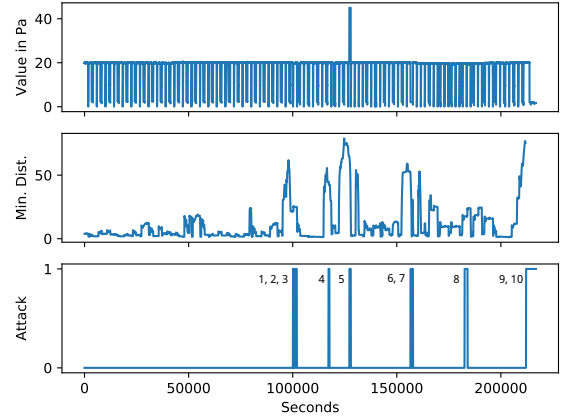

Figure 4: *Matrix Profile* of DPIT-301

sensor value. Attacks 1, 2, and 3 are detected in this process as well, similar to attacks 4 and 5, 6 and 7, as well as 9 and 10. Attack 8 is lower than noise in the minimal distance and thus not detectable. Since *Matrix Profiles* have been shown to perform well for such time series [6], especially in conjunction of different aspects of the process, they are evaluated for an extension in this work. Since *Matrix Profiles* calculate the minimal distance, an attack that occurs twice and has the same characteristic each time is not detected as an attack anymore, since it is already known behaviour. This means in praxis, an operator would have to detect every attack on the first try; in historical analysis, even this would not be possible. In order to create a more versatile approach, the instances of a motif are counted. An epsilon value is set to compare the current motif to all other motifs, all motifs whose distance is smaller than the epsilon value are added to a list. In doing so, the number of similar motifs can be extracted. Even if an attack occurs more than once and the masking of attack and normal behaviour results in the same pattern, the number of similar motifs is small, indicating a rare behaviour despite a low minimal distance. In addition to the standard *Matrix Profiles*, an extension has been implemented: In Figure 5, this information is added in the line *Similar Values*. This figure depicts data consisting of 85 000 instances from non-malicious data as well as about 60 000 instances of malicious data in which three blocks of attacks are expected. Attacks are indicated by a one in the corresponding line. It can be seen that the attack blocks are responsible for the highest minimal distances, as expected and already shown in Figure 4. Additionally, the similarities correspond to the *Matrix Profiles*. As a threshold for similarities, 20 was set. Even though there are two regions, around 18 000 and 32 000 milliseconds with zero similar values, the largest areas correspond to the attacks. Additionally, a time interval of 45 000 milliseconds has been analysed to get a better view of the behaviour, at the downside of fewer comparisons of motifs. This leads to a higher false positive rate, nicely shown for *LIT-301* in Figure 6. However, the same consideration of *DPIT-301* shows that the minimal distances are most prominent for the attacks, presented in Figure 7. The prominent length of zeros
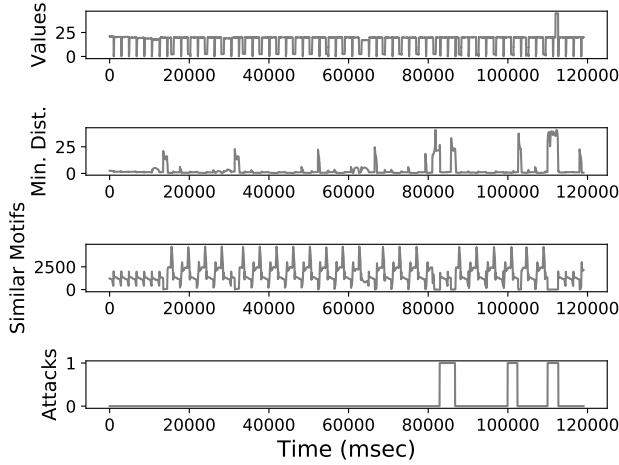
Figure 5: *Matrix Profile* and Similiarity Information of DPIT-301

Figure 7: *Matrix Profile* and Similiarity Information of DPIT-301 in a Smaller Interval

Table 1: Performance of One-Class Classifiers

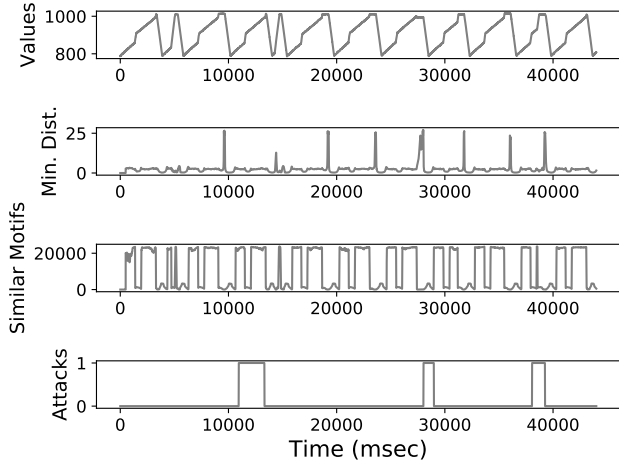| Preprocessing		OCSVM		Isolation Forest	
		Accuracy	F1-Score	Accuracy	F1-Score
bool	None	0.121 4	0.216 5	0.262 4	0.205 8
	0 mean	0.121 4	0.261 7	0.618 2	0.187 3
	linear	0.224 4	0.212 5	0.576 2	0.188 2
	PCA	0.360 0	0.199 5	0.551 4	0.193 3
non_bool	None	0.121 4	0.216 5	0.266 0	0.206 3
	0 mean	0.121 4	0.216 5	0.447 1	0.195 3
	linear	0.240 9	0.210 6	0.405 7	0.196 8
	PCA	0.351 1	0.199 3	0.405 7	0.199 2

Figure 6: *Matrix Profile* and Similiarity Information of LIT-301 in a Smaller Interval

for the similar motifs is presented as well. From both these features, information about anomalies due to attacks can be derived.

5.2 *Isolation Forest* and *OCSVM*

As described in Section 4, *OCSVM*s and *Isolation Forests* are one-class classifiers. That means they are trained on one class of event while being capable of classifying between two classes in operation. In this work, the normal operation of the data set was employed as a training data set. It contains 499 220 instances of 51 sensor values each, monitored during eleven days of operation. After training, the classifier was tested on the remaining four days of operation during which several attacks occurred. The results are listed in Table 1. In general, the performance in these experiments is comparably bad. In case of *Isolation Forest*, pre-processing is able to improve the classification performance in terms of accuracy while
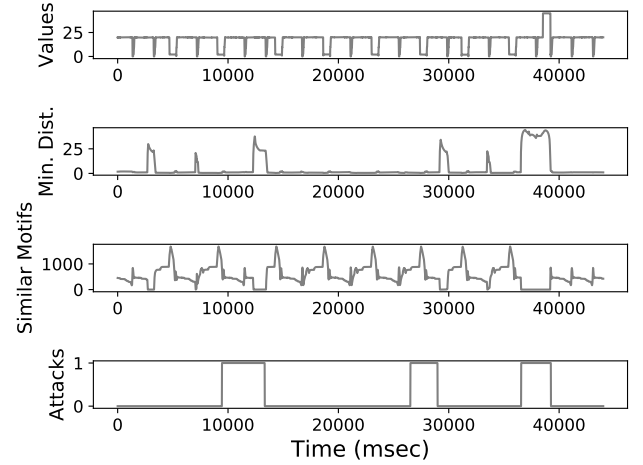
reducing the F1-score. In this case, applying *Principal Component Analysis* (*PCA*) procudes the best results. *PCA* is a method to map multidimensional features to a lower dimensional feature space with the most important features being the most prominent in the output vector [27]. The same goes for *OCSVM*, with a smaller improvement because of pre-processing and a lower classification quality overall. There are several hyper-parameters to be tuned. However, Changing ν and γ for *OCSVM* as well as changing the contamination factor or feature size for *Isolation Forest* result in any significant improvement. Furthermore, the data has been evaluated with and without taking boolean values into consideration, noted in Table 1 as *bool* and *non_bool* preliminary experiments indicate that ν as a metric for training errors and support vectors does not have much influence on the result. Additionally, scaling the data before training and testing leads to only positive classifications in preliminary tests with a reduced data set. Thus, all data in Table 1 are derived from the unprocessed data set. It should be noted that this classifier does not contain a notion of timing, in contrast to the *Matrix Profiles* and *LSTMs*.

6 CONCLUSION

The digitisation of industry creates the demand for an increase in industrial cyber security solutions. Due to legacy reasons, they should

integrate into existing applications. The time series-based approach *Matrix Profiles* is promising in terms of legacy-capabilities as well as detection. Furthermore, since little tuning of hyper-parameters is required, it is easy to set up and robust to different kinds of data. Furthermore, even though the detection capabilities are increased with a larger data base for comparison, no formal training is required. The addition to *Matrix Profiles*, presented in this work, can be used to detect attacks that occur multiple times and provide an increased level of security and traceability. One-class classifiers on the other hand require extensive training with a large amount of data. Despite pre-processing and tuning on hyper-parameters, they do not perform satisfactorily for the data evaluated in this work. Generally speaking, the adaption of novel techniques for intrusion detection is necessary in order to meet the current requirements. Combinations of anomaly detection methods with deception solutions, such as presented by *Fraunholz et al.* [10, 11]. Additionally, presenting the results in a fashion that is easily understandable for human operators, especially non-experts in cyber security is crucial for effective defense against attacks. *Lohfink et al.* present a visual representation of the results obtained in this work [22].

ACKNOWLEDGMENTS

REFERENCES

[1] 2017. Study on Communication for Automation in Vertical Domains. 3GPP TR 22.804, V1.0.0.
[2] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*, Vol. 2. Springer, Berlin, Heidelberg, 1–4. https://doi.org/10.1007/978-3-642-00296-0_5
[3] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92)*. New York, NY, USA, 144–152.
[4] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20, 3 (September 1995), 273–297.
[5] Simon Duque Anton, Daniel Fraunholz, Christoph Lipps, Frederic Pohl, Marc Zimmermann, and Hans Dieter Schotten. 2017. Two Decades of SCADA Exploitation: A Brief History. In *2017 IEEE Conference on Application, Information and Network Security (AINS)*. 98–104. https://doi.org/10.1109/AINS.2017.8270432
[6] Simon Duque Anton, Daniel Fraunholz, and Hans Dieter Schotten. 2019. Using Temporal and Topological Features for Intrusion Detection in Operational Networks. In *ARES '19: Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, ACM. https://doi.org/10.1145/3339252.3341476
[7] Simon Duque Anton, Alexander Hafner, and Hans Dieter Schotten. 2019. Devil in the Detail: Attack Scenarios in Industrial Applications. In *2019 IEEE Security and Privacy Workshops*. IEEE, IEEE.
[8] Simon Duque Anton, Sapna Sinha, and Hans Dieter Schotten. 2019. Anomaly-based Intrusion Detection in Industrial Data with SVM and Random Forests. In *Proceedings of the 27th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE.
[9] Cheng Feng, Tingting Li, and Deeph Chana. 2017. Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 261–272. https://doi.org/10.1109/DSN.2017.34
[10] Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. 2017. Investigation of Cyber Crime Conducted by Abusing Weak or Default Passwords with a Medium Interaction Honeypot. *International Conference On Cyber Security And Protection Of Digital Services* (2017).
[11] Daniel Fraunholz, Daniel Krohmer, Simon Duque Anton, and Hans Dieter Schotten. 2017. YAAS - On the Attribution of Honeypot Data. *International Journal on Cyber Situational Awareness* 2, 1 (2017), 31–48.
[12] Wei Gao and Thomas H. Morris. 2014. On Cyber Attacks and Signature Based Intrusion Detection for Modbus Based Industrial Control Systems. *Journal of Digital Forensics, Security and Law* 9, 1 (2014). https://doi.org/10.15394/jdfsl.2014.1162
[13] Alexander Giehl, Norbert Wiedermann, and Sven Plaga. 2019. A Framework to Assess Impacts of Cyber Attacks in Manufacturing. In *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering (ICCAE 2019)*. ACM, New York, NY, USA, 127–132. https://doi.org/10.1145/3313991.3314003
[14] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. 2016. A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *Proceedings of the 11th International Conference on Critical Information Infrastructures Security*.
[15] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. 2017. Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. 140–145. https://doi.org/10.1109/HASE.2017.36
[16] Vinay M. Igure, Sean A. Laughter, and Ronald D. Williams. 2006. Security issues in SCADA networks. *Computers & Security* 25 (2006), 498–506.
[17] iTrust Center for Research in Cyber Security. [n. d.]. S317 Dataset. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#s317
[18] iTrust Centre for Research in Cyber Security. 2018. *Secure Water Treatment (SWaT) Testbed*. Technical Report 4.2. Singapore University of Technology and Design.
[19] Eric D. Knapp and Joel Thomas Langill. 2014. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress.
[20] Robert D. Larkin, Juan Lopez Jr., Jonathan W. Butts, and Michael R. Grimaila. 2014. Evaluation of Security Solutions in the SCADA Environment. *SIGMIS Database* 45, 1 (March 2014), 38–53. https://doi.org/10.1145/2591056.2591060
[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining (ICDM)*. IEEE, 413–422.
[22] Anna-Pia Lohfink, Simon Duque Anton, Hans Dieter Schotten, Heike Leitte, and Christoph Garth. 2019. Security in Process: Visually Supported Triage Analysis in Industrial Process Data. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security 2019*. IEEE Symposium on Visualization for Cyber Security (VizSec-2019), October 20-25, Vancouver, British Columbia, Canada. IEEE, IEEE.
[23] Leandros A Maglaras and Jianmin Jiang. 2014. Intrusion Detection in SCADA Systems Using Machine Learning Techniques. In *2014 Science and Information Conference*. IEEE, 626–631.
[24] Modbus. 2012. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
[25] Modbus-IDA. 2006. MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf
[26] Abdullah Mueen, Suman Nath, and Jie Liu. 2010. Fast Approximate Correlation for Massive Time-series Data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)*. ACM, New York, NY, USA, 171–182. https://doi.org/10.1145/1807167.1807188
[27] Karl Pearson. 1901. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. https://doi.org/10.1080/14786440109462720
[28] Sven Plaga, Norbert Wiedermann, Simon Duque Anton, Stefan Tatschner, Hans Dieter Schotten, and Thomas Newe. 2019. Securing Future Decentralised Industrial IoT Infrastructures: Challenges and Free Open Source Solutions. *Future Generation Computer Systems* 93 (2019), 596–608.
[29] Sven Plaga, Norbert Wiedermann, Alexander Giehl, and Thomas Newe. 2018. Future Proofing IoT Embedded Platforms for Cryptographic Primitives Support. *2018 12th International Conference on Sensing Technology (ICST)*, 52–57. https://doi.org/10.1109/ICSensT.2018.8603610
[30] Peter Schneider and Konstantin Böttinger. 2018. High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks. In *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy (CPS-SPC '18)*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3264888.3264890
[31] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. 2000. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems*. 582–588.
[32] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. F. Wang. 2014. Multiattribute SCADA-Specific Intrusion Detection System for Power Networks. *IEEE Transactions on Power Delivery* 29, 3 (June 2014), 1092–1102. https://doi.org/10.1109/TPWRD.2014.2300099
[33] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, ABdullah Mueen, and Eamonn Keogh. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 1317–1322. https://doi.org/10.1109/ICDM.2016.0179