2018 IEEE International Conference on Data Mining Workshops (ICDMW)

Time is of the Essence: Machine Learning-based Intrusion Detection in Industrial Time Series Data

Simon Duque Anton, Lia Ahrens, Daniel Fraunholz, and Hans D. Schotten
Intelligent Networks Research Group
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
{simon, lia, daniel, hans_dieter}.{duque_anton, ahrens, fraunholz, schotten}@dfki.de

***Abstract*—The Industrial Internet of Things drastically increases connectivity of devices in industrial applications. In addition to the benefits in efficiency, scalability and ease of use, this creates novel attack surfaces. Historically, industrial networks and protocols do not contain means of security, such as authentication and encryption, that are made necessary by this development. Thus, industrial IT-security is needed. In this work, emulated industrial network data is transformed into a time series and analysed with three different algorithms. The data contains labeled attacks, so the performance can be evaluated. *Matrix Profiles* perform well with almost no parameterisation needed. *Seasonal Autoregressive Integrated Moving Average* performs well in the presence of noise, requiring parameterisation effort. *Long Short Term Memory*-based neural networks perform mediocre while requiring a high training- and parameterisation effort.**

***Index Terms*—Time Series Analysis, Matrix Profiles, Machine Learning, Mathematical Statistics, Industrial IT-Security**

I. Introduction

Over the last four and a half decades, automation and industrial control have been ever changing. Today, the introduction of the Internet of Things (IoT) into production, the so-called Industrial Internet of Things (IIoT), is emerging. With all the benefits arising from the IIoT, however, risks are evolving as well. Over the last two decades, an increase in attacks on energy and automation systems has been noted [1]. Many infamous examples have been discovered, such as *StuxNet*, *Industroyer* and *Black Energy*. Unfortunately, industrial Information Technology (IT) security has not evolved as fast as the IIoT. While home and office appliances always had to combat attacks, industrial applications have been deemed secure [2]. This paradigm is currently changing. Unlike home and office intrusion detection, however, security solutions are not as mature. Partly, this is due to the scarcity of data to test industrial Intrusion Detection System (IDS) applications on. As the communication patterns of industrial systems differ from home and office-based network traffic, the same IDS cannot be adapted for industrial application. In this work, three time series anomaly detection algorithms, *Matrix Profiles*, *Seasonal Autoregressive Integrated Moving Average* (*SARIMA*)- and *Long Short Term Memory* (*LSTM*)-approach are evaluated on an industrial data set based on the *Modbus/TCP* protocol, a common and open source communication protocol for industrial applications. The remainder of this work is structured as follows: Section II gives an an overview of time series-based intrusion detection, as well as industrial intrusion detection. After that, the data set is introduced in Section III. In Sections IV, V and VI, three time series-based algorithms for anomaly detection are applied to the data set and the results are discussed respectively. This work is concluded in Section VII.

This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Foerderkennzeichen KIS4ITS0001, IUNO). The authors alone are responsible for the content of the paper. An extended version of this work can be found at: https://arxiv.org/abs/1809.07500

II. Related Work

Supervisory Control And Data Acquisition (SCADA) has been identified as a likely and promising target for cyber attackers [3], [4]. Temporal properties as a feature for intrusion detection have been well-researched [5], [6]. Different works discuss wavelet analysis of network data represented as a time series [7], [8]. Furthermore, Wiener Filtering with *Autoregressive Moving Average* (*ARMA*) modeling is evaluated by *Celenk et al.* [9]. *K-means clustering* of time series data is performed [10], as well as statistical analysis of temporal distribution [11], [12]. Furthermore, neural networks have been applied to industrial network intrusion detection. *Filonov et al.* propose *LSTM*-based intrusion detection on a synthetic data set that has been generated in their work [13]. *Lin et al.* analyse the time distribution of synthetic and real SCADA network data and identify deviations [14]. *Linda et al.* create an intrusion detection system based on neural networks [15]. In addition to neural networks, other techniques have been employed to detect attacks in industrial networks. State-based intrusion detection was evaluated by *Goldenberg and Wool* [16], *Fovino et al.* [17] and *Carcano et al* [18]. Additionally, rule-, signature- and multiattribute-based intrusion detection systems have been evaluated [19]–[21]. Since Security Event and Information Management (SIEM)-systems are increasingly relevant for industrial intrusion detection, *Oman et al.* proposed a method to integrate intrusion detection into SIEM-systems for SCADA networks [22]. There are many works addressing the application of neural networks to intrusion detection for home and office networks [23], [24]. Unfortunately, these works lack standardised data sets for evaluation. Often, the *KDD Cup'99* data set [25] that has been proven to contain artifacts that lead to overfitting [26] is used. The data set we are analysing in

2375-9259/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDMW.2018.00008

this work [27] has been analysed on packet level by *Duque Anton et al.* [28].

III. Time Series Data Set for Intrusion Detection

Lemay and Fernandez created a batch of *Modbus/TCP* data sets of an emulated industrial application [27]. They implemented a physical simulation model of electronic circuit breakers. This physical model was connected via a *Modbus/TCP* connection. It consisted of three to twelve software Programmable Logic Controllers (PLCs) that were queried periodically by one or two Master Terminal Units (MTUs). Furthermore, aperiodic user behaviour was introduced as queries. After the traffic was recorded, exploits were introduced. These exploits were generated with the penetration testing tool *metasploit* [29] and are based on the TCP/IP layer. Unfortunately, no *Modbus* protocol-specific exploits have been performed. However, *Lemay and Fernandez* proposed a different batch of data where they used the lowest bits in the *Modbus* payload as a covert channel [27]. In this work, three different data sets have been evaluated: *ds1* is called “Moving_two_files_Modbus_6RTU”, *ds2* is called “Send_a_fake_command_Modbus_6RTU_with_operate” and *ds3* is called “CnC_uploading_exe_modbus_6RTU_with_operate” in [27]. These data sets and their characteristics are listed in Table I. In this table, the number of packets in total is listed, the length of communication, as well as the number of malicious packets, the number of attacks defined as sequences of malicious packets. All data sets have polling intervals of 10 seconds. In contrast to *ds2* and *ds3*, *ds1* does not contain human interaction. In this work, each second of network traffic has been aggregated as one data point, also called event. Features that are collected and evaluated are, among others, the number of protocols detected, the number of packets and bytes, the flags and function codes in a one-hot encoding. Due to preliminary analysis, three features are employed in the anomaly detection: the number of packets per second, the number of port pairs per second and the number of IP pairs per second.

TABLE I: Characteristics of Analysed Data Sets

Name	# of packets	Length (s)	# of mal. packets	# of attacks
ds1	3,319	190	75	4
ds2	11,166	670	10	1
ds3	1,426	70	121	2

IV. Matrix Profiles

Matrix Profiles are employed in order to detect time series-based anomalies in the data sets in this section. First, he algorithm is introduced. After that, the three data sets introduced above are evaluated. Finally, the performance of *Matrix Profiles* is discussed.

The *Matrix Profile* algorithm is a method to calculate similarities in time series. It has been introduced by *Yeh et al.* in 2016 [30]. A sequence from a time series is compared to every other sequence of the same length within the time series. The distances are calculated and stored. This distance is a metric for similarity. If a sequence has a low minimal distance, a sequence with a related characteristic is present in the time series. If the minimal distance of a sequence is relatively high, it is unique in the time series. This property is suited to find outliers that can indicate attacks. To obtain the *Matrix Profile*, each windowed sequence of length m is compared to each other sequence of length m in the time series. An interval of $\frac{m}{2}$ before and after the start of the sequence under observation is excluded, as this would result in a trivial match. A sequence x_i has a distance of 0 from itself. After the distances are calculated, they are stored in a matrix. The minimum of of each column is stored, indicating the minimal distance of the given sequence from any other sequence. As the concept of distance is not easily mappable to formal metrics for classifier quality, no classic metric for classifier quality is employed in evaluating *Matrix Profiles*. Due to the length of the sliding window, the raise in distance is longer than the attack itself. This would falsify a metric in creating a large amount of false positives. Instead, a perfect threshold is calculated in a fashion that it is minimal, while still able to identify every attack.

A. Evaluation of ds1

The *Matrix Profile* of *ds1* is depicted in Figure 1. The curve describes the minimal distance of a sequence to any other, previously occuring, sequence. As shown in Table I, *ds1* contains four attacks in total, depicted by the series of dark gray dots. The attacks can be clearly distinguished by comparing the distance values to an ideal threshold, which is displayed by a solid line. Any feature is capable of indicating the attacks. The four increases in distance map to the beginning of the attacks. Only the first attack is detected with one second delay. This is due to the fact that it falls into a polling request perfectly, hardly altering the expected behaviour. Its second event, however, clearly indicates an attack. The increase of distance on the very end of the curves in Figure 1 is an artifact due to the data formatting.

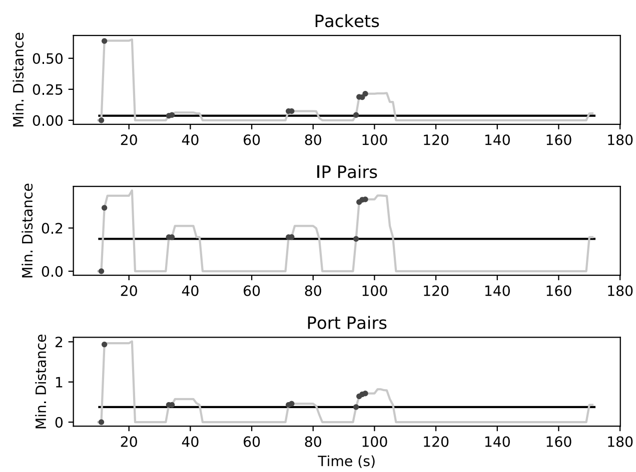

Fig. 1: *Matrix Profile* for *ds1*

B. Evaluation of ds2

As shown in Figure 2, *ds2* contains a lot of aperiodic traffic indicating anomalous events. Only one of them is malicious, marked by the dark gray dot. In this case, the number of packets per second is the most reliable indicator for attacks. It would generate only one false positive if an appropriate threshold was used, as indicated by the solid line. The other two features create more false positives. Despite the noise, a relatively good detection of the attack is possible.

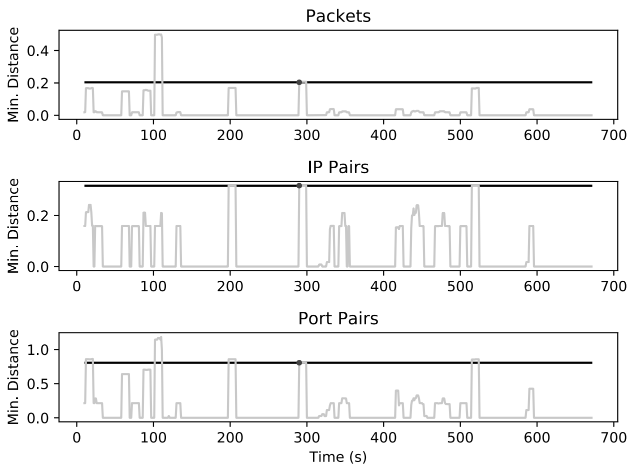

Fig. 2: *Matrix Profile* for *ds2*

C. Evaluation of ds3

Considering the fact that *ds3* contains manual, aperiodic operations, the *Matrix Profile* approach works exceedingly well. The distances are shown in Figure 3. Especially the numbers of packets and port pairs are capable of identifying attacks that are indicated by dark gray dots, if the perfect threshold, illustrated by the solid line, is employed. As discussed before, the rise in distance has a longer duration due to the length of the sliding window and the influence of anomalous events on the window. Still, *Matrix Profiles* are relatively robust to this, and the location of an attack can be detected easily as it is the first instance of a raised distance.

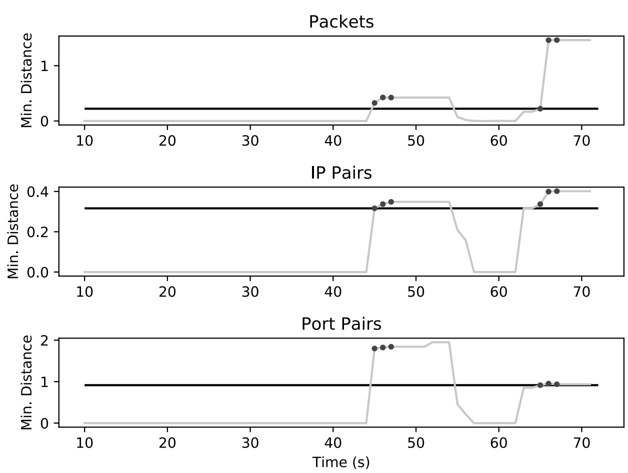

Fig. 3: *Matrix Profile* for *ds3*

D. Discussion

Matrix Profiles are well suited to detect anomalies in data with periodic characteristics while tolerating a certain amount of noise. The predominant benefit is the ease of use as there is only one hyperparameter - m - to be defined. The algorithm is robust to changes in m so that fine-tuning is rarely necessary. Furthermore, efficient implementations of the distance calculation allow for in-time calculation of distances and therefore for real-time discovery of attacks. The choice of a threshold depends on the characteristics of a data set.

V. *SARIMA* APPROACH

In this section we carry out a *SARIMA* approach for properly modelling and forecasting short-term future values of time series extracted from regular network traffic with periodical characteristics. Network intrusion can be identified by capturing data points that vary enough from the prediction.

A. Seasonal ARIMA-processes

A stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ is called seasonal ARIMA-process with period s and denoted by $SARIMA(p,d,q) \times (P,D,Q)_s$ if $\{Y_t\}_{t\in\mathbb{Z}}$ with

$$Y_t = (1-U^{-1})^d(1-U^{-s})^D X_t \quad \text{for } t \in \mathbb{Z} \tag{1}$$

is a stationary autoregressive-moving average ($ARMA$-) process of the form

$$A(U^{-1})F(U^{-s})Y_t = D(U^{-1})G(U^{-s})\epsilon_t, \quad t \in \mathbb{Z}, \tag{2}$$

where $\{\epsilon_t\}_{t\in\mathbb{Z}}$ is the innovation process which is supposed to be white noise, i.e. a series of uncorrelated random variables with zero mean and finite variance σ_ϵ^2, U denotes the shift operator, i.e., $U : X_t \mapsto X_{t+1}$, and A, F, D, G refer to the characteristic polynomials related to the $ARMA$-process, i.e.

$$\begin{aligned} A(z) &= 1 - \sum_{k=1}^{p} \alpha_k z^k, \quad D(z) = 1 + \sum_{k=1}^{q} \theta_k z^k, \\ F(z) &= 1 - \sum_{k=1}^{P} \phi_k z^k, \quad G(z) = 1 + \sum_{k=1}^{Q} \gamma_k z^k \end{aligned} \tag{3}$$

([31, Chap. 9.1.3]). We assume for simplicity that $\{\epsilon_t\}_{t\in\mathbb{Z}}$ is Gaussian white noise, i.e., $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$ for all $t \in \mathbb{Z}$.

B. Data adaptive model choice

In order to apply asymptotic results from mathematical statistics, it is reasonable to use the data without intrusion from the largest dataset (i.e.,with the longest duration) *ds2* to model regular traffic. This also enables us to highlight the benefit of using the *SARIMA* approach for noisy data. The only intrusion in the aforementioned data set is injected during the time interval from the 289th to the 290th second without any effect on the subsequent traffic. We therefore choose the time series of data points obeserved from the 300th second to the 670th second of that data set for modelling the normal network behavior (training data). We then apply our forecast model to the remainder of available data (test data). It turns out that in one-dimensional case, using the number of active port pairs for modeling and testing yields the most accurate result on intrusion detection. In the sequel, the time series of number of active port pairs per second extracted from *ds2* is denoted by $\{X'_t\}_{t=1}^N$ with $N = 370$ (s).

First, aiming to confirm the periodical character induced by the 10-seconds polling interval, we start by analysing the autocorrelogram of the time series $\{X'_t\}_{t=1}^N$, which is the plot of sample autocorrelations expressed in terms of

$$\hat{\rho}_\tau^{X'} = \frac{\hat{r}_\tau^{X'}}{\hat{r}_0^{X'}} \quad \text{with} \quad \hat{r}_\tau^{X'} = \frac{1}{N}\sum_{k=1}^{N-\tau}(X'_k - \bar{X}'_N)(X'_{\tau+k} - \bar{X}'_N) \tag{4}$$

against lags $\tau = 0, 1, ..., N-1$. The sample autocorrelation oscillates with constant frequency of ten seconds, which indicates the existence of a periodical component with $s = 10$ in (2). In order to remove the heteroscedasticity of data in respect of state dependent variance and motivated by regression models with time series error terms [31, Chap. 9.5.1], we transform the original time series into a seasonally centered one $\{X_t\}_{t=1}^N$, that is,

$$X_{j+us} := X'_{j+us} - \frac{1}{N/s}\sum_{i=0}^{N/s-1} X'_{j+is}, \tag{5}$$

for $j = 1, \ldots, s-1$, $u = 0, \ldots, N/s-1$. Since our time series is not expected to have any trend component, we choose $d = D = 0$ in (1), thereby obtaining a $SARIMA(p, 0, q) \times (P, 0, Q)_{10}$ model with the time series $\{Y_t\}_{t=1}^N$ given by $Y_t := X_t$ for all $t = 1, \ldots, N$. (Note that $\bar{Y}_N = \bar{X}_N = 0$.)

In order to determine the orders of the moving average part (Q, q) and the autoregressive part (P, p), we analyse the autocorrelogram and partial autocorrelogram related to $\{Y_t\}_{t=1}^N$, respectively. The sample autocorrelations $\hat{\rho}_\tau^Y$ for $\tau = 0, 1, ..., N-1$ can be computed as in (4), whereas the sample partial correlations $\hat{\pi}_\tau^Y$ can be determined by means of the Yule-Walker equation, which gives $\hat{\pi}_\tau^Y = (\hat{R}_\tau^Y)^{-1}\hat{r}^Y(\tau)$ with $\hat{R}_\tau^Y = (\hat{r}_{t-k}^Y)_{1\le t,k\le\tau}$, $\hat{r}^Y(\tau) = (\hat{r}_1^Y, ...\hat{r}_\tau^Y)^T$. We then obtain possible choices of P and Q by considering $\hat{\pi}_{ks}^Y$ and $\hat{\rho}_{ks}^Y$ for $k \ge 0$ and picking those values of k where $\hat{\pi}_{(k+1)s}^Y$ and $\hat{\rho}_{(k+1)s}^Y$ begin to fall exponentially towards zero, respectively. Moreover, analysing $\hat{\pi}_1^Y, ..., \hat{\pi}_{s-1}^Y$ and $\hat{\rho}_1^Y, ..., \hat{\rho}_{s-1}^Y$ in the above manner provides possible choices of p and q, respectively. The choice of P, Q, p, q is not unique in general; a final decision among the possible candidates can be made, for instance, by means of the Akaike information criterion. Since our focus is not on delivering the best possible forecast but on detecting outliers and as these can also occur in early periods (cf. *ds1*), also doing so with the least possible amount of preceding data, we make a compromise of retaining accuracy and computational simplicity and accept the combination of orders $(4, 0) \times (1, 0)$. Summing up, the above consideration leads to our choice of the model $SARIMA(4, 0, 0) \times (1, 0, 0)_{10}$ for $\{Y_t\}_{t=1}^N$.

C. Parameter estimation

Since $\{\epsilon_t\}_t$ is assumed to be Gaussian white noise, we use the least squares estimation which approximately delivers the asymptotically efficient maximum likelihood estimate of the parameters α, ϕ and σ_ϵ. The functional we are going to minimize reads as $f(\alpha, \phi) := \sum_{t=Ps+p+1}^N \epsilon_t(\alpha, \phi)^2$ with

$$Y_t^*(\alpha, \phi) = \sum_{j=1}^p \alpha_j Y_{t-j} + \sum_{k=1}^P \phi_k Y_{t-sk} - \sum_{j=1}^p\sum_{k=1}^P \alpha_j\phi_k Y_{t-sk-j}, \tag{6}$$

$$\epsilon_t(\alpha, \phi)_t = Y_t - Y_t^*(\alpha, \phi), \quad t \ge Ps + p + 1. \tag{7}$$

It holds for the least squares estimate that

$$(\hat{\alpha}, \hat{\phi}) = \underset{\alpha\in\mathbb{R}^p, \phi\in\mathbb{R}^P}{\arg\min} f(\alpha, \phi), \quad \hat{\sigma}_\epsilon^2 = MSE = \frac{f(\hat{\alpha}, \hat{\phi})}{N - Ps - p}.$$

We solve the minimization problem numerically by means of a gradient descent procedure. Then the one-step prediction of $\{Y_t\}_t$ and the residuals $\{\epsilon_t(\hat{\alpha}, \hat{\phi})\}_t$ are obtained by (6, 7).

In a final step, let us verify that indeed our model is reasonable in the sense that the residuals $\epsilon_t(\hat{\alpha}, \hat{\phi})$ for $t \ge Ps + p + 1$ are white noise. To this end, we consider their sample autocorrelations $\hat{\rho}_1^\epsilon, \hat{\rho}_2^\epsilon, \ldots$ and conduct the *Ljung-Box* test [31, Chap. 8.2.2] that uses as test statistic $Q := N(N+2)\sum_{k=1}^H (\hat{\rho}_k^\epsilon)^2/(N-k)$, where we choose $H = \lfloor 2\sqrt{N} \rfloor$ and critical region $\{Q > q_{1-\alpha}(\chi^2_{H-(Ps+p)})\}$ for a significance level of $\alpha = 0.05$. It turns out that in our setting, $Q = 33.40707$, $q_{1-\alpha}(\chi^2_{H-(Ps+p)}) = 36.9982$.

D. Intrusion detection

For each test data set, we first transform the corresponding time series $\{Z'_t\}_t$ into a seasonally centred one $\{Z_t\}_t$ in the sense of (5) with fixed sample seasonal means obtained from the *training* data $\{X'_t\}_t$. Then we apply the model from V-B, V-C to the transformed test time series $\{Z_t\}_t$ and compute the one-step prediction errors e_t^Z in terms of (6, 7) for $t \ge Ps + p + 1$. Due to the different nature of the three test data sets in respect of manual operations, which belong to non-intrusion anomalies, we set individual thresholds for evaluating prediction errors in different data sets. As soon as the absolute prediction error $|e_{t_0}^Z|$ exceeds the threshold at some time t_0, the traffic related to that state Z_{t_0} in the time series is classified as an intrusion. In order to prevent consequent errors, we also remove the detected attack traffic from the data immediately after detection, i.e., we replace the detected outlier Z_{t_0} by the corresponding regular value before we continue the detection procedure for $t > t_0$.

The final results of the above approach are summarised in Table II. It turns out that, by chosing the proper threshold for each data set, all attack traffic can be accurately detected within one second while producing only a single false positive.

[1] As this anomaly occurres before $Ps+p$ seconds, the model is not applicable here.

TABLE II: Results of SARIMA based Anomaly Detection

Dataset	Threshold	Attack (s)	Detection Time (s)
ds2	$3q_{0.9995}(\mathcal{N}(0,\hat{\sigma}_\epsilon^2))$	289.4079	290
ds1	$q_{0.9995}(\mathcal{N}(0,\hat{\sigma}_\epsilon^2))$	10.8980	N/A[1]
		32.9679	34
		71.5955	72
		93.6086	94
ds3	$q_{0.9995}(\mathcal{N}(0,\hat{\sigma}_\epsilon^2))$	44.3293	45
		—	63
		64.1758	65

E. Discussion

Overall, the main advantage of the *SARIMA* approach is that it is still powerful in the presence of noise. It also provides a parsimonious presentation of time series with periodical characteristics, using few parameters. However, it requires individual model adjustment each time dealing with a new time series.

VI. Long Short Term Memory

LSTM is a kind of neural network proposed by *Hochreiter and Schmidhuber* in 1997 to overcome the vanishing gradient problem [32]. This problem occurs when long-term dependencies are not considered accordingly by a recurrent neural network. The network "forgets" the events and cannot correlate dependencies.

A. Introduction to LSTM

LSTM is a kind of neural network designed to keep information over long periods of time. Due to this ability, *LSTM* networks need the ability to reset parameters. This can be done with forget gates [33]. *LSTM* networks consist of cells that are interconnected. Four different parameterisations have been employed in this work: The length of input sequences has been set to 10 and 20, the number of layers has been set to 1 and 3. Furthermore, 400 neurons have been used. The training has been performed on the *ds1* that has been stripped of malicious events using 20,000 iterations with a learning rate of 0.001 and batch sizes of 50. The *LSTM* networks have been used as predictors. They predicted the next event which was then compared to the actual event. In order to prevent consequent errors, the detected attack traffic was removed from the data immediately after detection before continuing the prediction process. All prediction errors were calculated as the absolute difference of predicted and actual value. After that, two threshold values were calculated. The first threshold was the minimal value, so that all malicious evens are above the threshold (*MA*). The second threshold was chosen so that all non-malicious events were below the threshold value (*NM*). Only the best performing set ups are presented in the following.

B. Evaluation of LSTM

After setting the threshold as discussed in the previous subsection, the F1-score as well as the accuracy are calculated. These metrics were evaluated for each feature, packet count (*PC*), IP pairs (*IP*) and port pairs (*PP*) respectively. The performance of *LSTM* networks is presented in Table III. Table III shows that the accuracy is always higher than the F1-score. This is due to the fact that t_p as well as t_n are considered in calculating the accuracy, while the F1-score puts an emphasis on t_p. As shown in Table I, the malicious events are magnitudes smaller than the non-malicious ones. This is a common problem in anomaly detection, as an anomaly happens less frequently [34]. It leads to a negligible t_p on the accuracy, most prevalently shown with *ds2*: The only attack was considered as non-malicious and thus a f_n, but the accuracy appears to be almost perfect. Nevertheless, *LSTM* performed relatively good analysing *ds1* and *ds3* considering some features.

VII. Conclusion

In this work, three algorithms for time series-based anomaly detection were evaluated on a synthetic data set containing network traffic of an industrial use case. They were analysed with respect to their capability in detecting attacks that were introduced to that data set, as well as the effort required to parameterise and train the algorithms. The *Matrix Profile* approach performs very well in comparison. Most attacks can be found with few false positives. Furthermore, the parametrisation effort requires only one hyperparameter that is robust to change. Additionally, the training data set is small. The *SARIMA* approach provides high forecasting accuracy and detection performance with few parameters. The theory of time series analysis from mathematical statistics provides a clearly defined procedure for the data adaptive model choice and the model adequacy check, e.g. the *Ljung-Box* test. Despite the high computational effort for training a large number of parameters, the performance of *LSTM* network is less convincing than that of the other two algorithms. It strongly depends on the nature of the data. In summary, the time series-based anomaly detection methods discussed in this work are effective in detecting cyber attacks in industrial network traffic. Their efficiency could be improved by incorporating context information [35] such as authentication when operating manually, and sensible aggregation of this information [36].

References

[1] S. Duque Anton, D. Fraunholz, C. Lipps, F. Pohl, M. Zimmermann, and H. D. Schotten, "Two decades of scada exploitation: A brief history," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, November 2017, pp. 98–104.

[2] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," *Computers & Security*, vol. 25, pp. 498–506, 2006.

[3] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, ser. ITHINGSCPSCOM. Washington, DC, USA: IEEE Computer Society, 2011, pp. 380–388.

[4] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–29, Mar. 2014.

[5] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, September 2014.

TABLE III: Performance of *LSTM*

Feature		*ds1*		*ds2*		*ds3*	
		MA (%)	*NM (%)*	*MA (%)*	*NM (%)*	*MA (%)*	*NM (%)*
PC	*Accuracy*	98.4293	97.9058	99.5529	99.8510	95.7746	95.7746
	F1-Score	86.9565	75.0000	40.0000	0	80.0000	66.6667
IP	*Accuracy*	90.0524	98.9529	99.9529	99.8510	98.5915	92.3077
	F1-Score	51.2821	88.8889	66.6667	0	92.3077	28.5714
PP	*Accuracy*	98.9529	99.4764	99.1058	99.8510	98.5915	92.9577
	F1-Score	90.9091	94.7368	25.0000	0	92.3077	28.5714

[6] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 343–356, March 2010.

[7] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP J. Adv. Signal Process*, vol. 2009, pp. 1–16, January 2009.

[8] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 71–82.

[9] M. Celenk, T. Conley, J. Graham, and J. Willis, "Anomaly prediction in network traffic using adaptive wiener filtering and arma modeling," in *2008 IEEE International Conference on Systems, Man and Cybernetics*, October 2008, pp. 3548–3553.

[10] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using kmeans clustering," in *In GI/ITG Workshop MMBnet*, 2007.

[11] M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. W. Hong, "A flow-based method for abnormal network traffic detection," in *2004 IEEE/IFIP Network Operations and Management Symposium*, vol. 1, April 2004, pp. 599–612.

[12] A. Sperotto, R. Sadre, and A. Pras, "Anomaly characterization in flow-based traffic time series," in *Proceedings of the 8th IEEE International Workshop on IP Operations and Management*, ser. IPOM '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 15–27.

[13] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model," *CoRR*, 2016.

[14] C.-Y. Lin, S. Nadjim-Tehrani, and M. Asplund, "Timing-based anomaly detection in SCADA networks," 2017. [Online]. Available: https://pdfs.semanticscholar.org/bc74/ca2e548c1567f4bd8794d480b83a91115f32.pdf

[15] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *2009 International Joint Conference on Neural Networks*, June 2009, pp. 1827–1834.

[16] N. Goldenberg and A. Wool, "Accurate modeling of Modbus/TCP for intrusion detection in scada systems," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 63 – 75, 2013.

[17] I. N. Fovino, A. Carcano, T. De Lacheze Murel, A. Trombetta, and M. Masera, "Modbus/DNP3 state-based intrusion detection system," in *24th IEEE International Conference on Advanced Information Networking and Applications(AINA)*, April 2010, pp. 729–736.

[18] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, "State-based network intrusion detection systems for SCADA protocols: A proof of concept," in *CRITIS 2009: Critical Information Infrastructures Security*, vol. 6027. Springer, Berlin, Heidelberg, June 2009, pp. 138–150.

[19] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. F. Wang, "Multiattribute scada-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092–1102, June 2014.

[20] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer, "Adapting bro into scada: Building a specification-based intrusion detection system for the dnp3 protocol," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, ser. CSIIRW '13. New York, NY, USA: ACM, 2013, pp. 5:1–5:4.

[21] J. Verba and M. Milvich, "Idaho national laboratory supervisory control and data acquisition intrusion detection system (scada ids)," in *2008 IEEE Conference on Technologies for Homeland Security*, May 2008, pp. 469–473.

[22] P. Oman and M. Phillips, "Intrusion detection and event monitoring in scada networks," in *Critical Infrastructure Protection*. Boston, MA: Springer US, 2008, pp. 161–173.

[23] R. C. Staudemeyer, "An overview of ip flow-based intrusion detection," *South African Computer Journal*, no. 56, pp. 136–154, July 2015.

[24] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, "Collective anomaly detection based on long short-term memory recurrent neural networks," in *Future Data and Security Engineering*, vol. 10018. Springer, October 2016, pp. 141–152.

[25] I. U. University of California. (1999) KDD cup 1999 data. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, July 2009, pp. 1–6.

[27] A. Lemay and J. M. Fernandez, "Providing scada network data sets for intrusion detection research," in *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*, Austin, TX, 2016.

[28] S. Duque Antn, S. Kanoor, D. Fraunholz, and H. D. Schotten, "Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set," in *Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES)*. ACM, 2018.

[29] Rapid7. metasploit. [Online]. Available: https://www.metasploit.com/

[30] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "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)*, December 2016, pp. 1317–1322.

[31] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis – Forecasting and Control*, 4th ed., ser. WILEY SERIES IN PROBABILITY AND STATISTICS. John Wiley & Sons, Inc., Hoboken, New Jersey, 2013.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, November 1997.

[33] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, pp. 2451–2471, 1999.

[34] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, July 2009.

[35] S. Duque Anton, D. Fraunholz, S. Teuber, and H. D. Schotten, "A question of context: Enhancing intrusion detection by providing context information," in *13th Conference of Telecommunication, Media and Internet Techno-Economics (CTTE-17)*, 2017.

[36] S. Duque Anton, D. Fraunholz, J. Zemitis, F. Pohl, and H. D. Schotten, "Highly scalable and flexible model for effective aggregation of context-based data in generic iiot scenarios," in *9th Central European Workshop on Services and their Composition (ZEUS-2017), February 13-14, Lugano, Switzerland*, April 2017, pp. 51–58.