An Improved Industrial Control System Device Logs Processing Method for Process-Based Anomaly Detection

Mukhtar Hussain*†, Ernest Foo*‡, Suriadi Suriadi*
*School of Electrical Engineering and Computer Science, Queensland University of Technonolgy, Australia
†Department of Electrical Engineering, COMSATS University Islamabad (Lahore Campus), Pakistan
‡School of Information and Communication Engineering, Griffith University, Australia
E-mails: {*m5.hussain, e.foo, s.suriadi*}*@qut.edu.au*

***Abstract*—Detecting process-based attacks on industrial control systems (ICS) is challenging. These cyber-attacks are designed to disrupt the industrial process by changing the state of a system, while keeping the system's behaviour close to the expected behaviour. Such anomalous behaviour can be effectively detected by an event-driven approach. Petri Net (PN) model identification has proved to be an effective method for event-driven system analysis and anomaly detection. However, PN identification-based anomaly detection methods require ICS device logs to be converted into event logs (sequence of events). Therefore, in this paper we present a formalised method for pre-processing and transforming ICS device logs into event logs. The proposed approach outperforms the previous methods of device logs processing in terms of anomaly detection. We have demonstrated the results using two published datasets.**

***Index Terms*—industrial control systems; intrusion detection system; system identification; process-related attacks**

I. Introduction

Industrial Control Systems (ICS) suffer from special category of cyber-attacks known as process-based attacks. These cyber-attacks are meant to disrupt the process or damage the critical infrastructure by changing the physical state of a system e.g., Stuxnet, BlackEnergy, and Triton etc. [1]. Intrusion detection can be a viable approach to safeguard the ICS against cyber-attacks [2]. Apart from monitoring the network traffic, there is growing interest in exploiting physical properties (e.g., temperature, pressure, etc.) of ICS in order to design an effective intrusion detection system (IDS) [3]. However, detecting process-based attacks is challenging. A stealthy attacker may try to disrupt the industrial process by changing the state of a system such that it does not to raise a safety alarm but does disrupt the process [4]. Fig 1 shows an example of process related attack where attacker "turned off the water pump before tank was full" and then tried to damage the water tank system by rapid "on and off" control of the water pump. While, the water tank level remains within the normal range i.e., it neither overflows nor underflows.

Process-based anomalies can be detected effectively by analysing event-driven data compared to statistical analysis of sampled data [5]. An event is referred to as instantaneous change in the state of a system such "pump is turned off". Petri net (PN) is a well known formalism for modelling event driven systems. Moreover, a PN model of system behaviour can been used for anomaly detection as a one-class classifier [6]. However, in ICS event-driven data or *event logs* are not readily available for PN *identification*. Events information is extracted from the ICS device logs [7]. Device logs contain record of device status with timestamps [8], as shown in Table. I. While, existing ICS device logs processing methods such as discussed in [7], [9] are limited to discrete event systems (DES) i.e., inputs and outputs of a system are logical (0/1 or on/off). However, most of the real-time automated systems are hybrid. For example, HVAC systems where input such as room temperature is a continuous valued dataset, while output sent by the controller is a logical valued dataset i.e., *turn-on* or *turn-off* the air-conditioning [10].

Therefore, the contribution for this paper is to present a formalised method to process the ICS device logs of *hybrid systems* to extract *event* information. A simple approach could be to rely solely on expert knowledge to convert device logs into *events*. However, relying on expert knowledge can be prone to errors [5]. One of the reason for that is, ICS undergo various changes (e.g., maintenance, replacement of faulty equipment, or software upgrade) during the lifespan and system may not work exactly the same as it was designed [11]. While, from the signal processing perspective quantising the real number data such as water-level of tank into logical states "Hi" or "Low" loses too much information [12].

In this paper we outline a formalised approach for transforming the ICS device logs into event logs based on change point analysis. We have demonstrated that proposed device log processing method shows superior performance for anomaly detection using PN model identification. It is evaluated using two different datasets, QUT's SIEMENS S7 dataset [7] and SUTD water treatment system [13]. The remainder of the paper is outlined as follows: In Section II we summarise the system identification and process mining algorithms. In Section III we define our event logs preprocessing method. In Section V, we evaluate our anomaly detection system. Finally, we conclude the paper in Section VI.

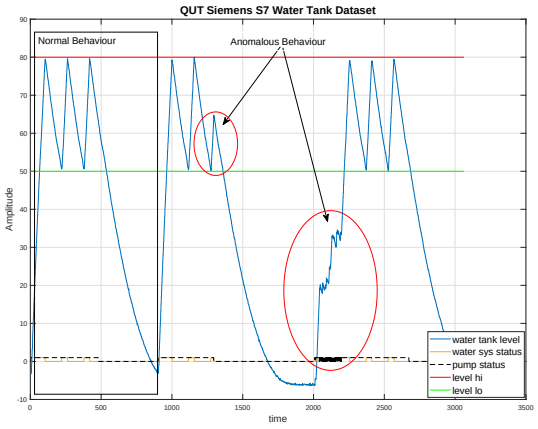

Fig. 1. Example of process related attack

II. Background & Related Work

In this section we provide a brief overview of anomaly detection for industrial control systems to address the question of why PN based anomaly detection can be effective. Moreover, an overview of the limitations of exiting ICS device logs processing for PN model generation is provided.

A. Anomaly Detection

Fauri *et al.* [4] presented feature selection based anomaly detection system, where each feature is similar to an event. However, features are extracted from the ICS device logs based on the expert knowledge, which may be prone to error as discussed earlier. Lin *et al.* [14] presented a timed *automata* based process oriented anomaly detection system. Automata have limited modelling power compared to Petri net and lack the ability to model concurrent behaviour. While, machine learning based algorithms such as artificial neural network lacks in providing actionable alerts [15].

Allen and Tilbury presented a Petri net model identification based anomaly detection system [5]. However, the proposed approach assumed that sequence of events is already known. Myers *et al.* [7] presented a process mining based anomaly detection system, where a Petri net model of a systems behaviour is extracted from device logs for anomaly detection. The proposed method provides an automated approach for processing of device logs into event logs that is required for anomaly detection using *process mining* algorithms. However, this approach is limited to device logs with logical values, while real valued device logs were ignored. Ignoring real valued parameters leads to a sub-optimal model discovery which causes a high false negative rate.

B. Device Logs Processing

In an industrial control system, programmable logic controllers (PLCs) automate the process by reading the input from sensors and sending output to actuators. The decision to change the state of actuator (output event) is made based on the set-points (input events). However, ICS device logs record the status of devices at specific time interval [8]. A snippet of ICS device logs is shown in Table. I.

TABLE I
A snippet of ICS device logs of Water Tank System in QUT SCADA Lab [7]

VarName	Timestamps	VarValue
HMI_Tank_Master_Mode	16/06/2017 5:41:08 PM	0
Tank_Level	16/06/2017 5:41:08 PM	-8.16274
Tank_Off_SP_Int	16/06/2017 5:41:08 PM	80
Tank_On_SP_Int	16/06/2017 5:41:08 PM	50
Tank_Pump_In_Auto	16/06/2017 5:41:08 PM	0
Tank_Pump_In_Manual	16/06/2017 5:41:08 PM	0
Tank_Pump	16/06/2017 5:41:08 PM	0
Tank_Read_Tank_Level	16/06/2017 5:41:08 PM	-8.16274

Device logs need to be converted into event logs for PN model generation. One may argue that ICS should be simply re-programmed to record events. However, it increases the initial cost of implementation limits its adaptability of many theoretical approaches in industrial environment [9]. On the other hand relying solely on expert's knowledge is prone to error. Moreover, the motivation behind PN model *identification* is to learn "how a system behaves?" rather than "how an expert think it behaves?" [5]. While, existing methods [7], [9] for the conversion of ICS device logs (I/O signal vectors) into events are limited to discrete event systems as discussed earlier.

To elaborate how existing event logs pre-processing methods leads to sub-optimal solution we have selected an example from [7]. Consider the water tank system where in normal conditions, a PLC maintains the tank level between low (50) and high (80) set-points by turning pump on and off shown in Fig. 1. An attacker sends a malicious command to the PLC and tries to turn off the pump before it reaches "tank level hi". Here all the output events i.e., turn off pump and the tank system are as in normal behaviour. However, it is the input event (tank level *high* and *lo*) in relation with the output events that can lead to the detection of anomalous behaviour. That's why these attacks remained undetected in [7].

The scope of this paper is limited to the processing of ICS device logs for Petri net model-based anomaly detection. Therefore, PN model identification and conformance checking algorithms are not discussed. Readers are referred to [7], [16] for a comprehensive survey on PN model discovery and anomaly detection.

III. New Device Log Processing Method

The significance of our methodology is to convert device logs of a hybrid control system into event logs that can be used for PN model discovery and conformance checking methods discussed in [16], [17]. An event log is a three-tuple $\mathfrak{L}_E = (\mathfrak{T}_E, \mathfrak{E}_{name}, \mathfrak{C}_{id})$ where $\mathfrak{T}_E = \langle t_1, t_2, \ldots, t_K \rangle$ is a sequence of timestamps for the sequence of events $\mathfrak{E}_{name} = \langle e_{name\_1}, e_{name\_2}, \ldots, e_{name\_K} \rangle$. While, $\mathfrak{C}_{id} = \{c_{id\_1}, c_{id\_2}, \ldots, c_{id\_j}\}$ is a finite set of integers an identifier for each process trace. A process trace is a group of events recorded during the execution of a process. While, timestamps provides the ordering of events in a process trace.

TABLE II
ICS DEVICE LOGS OF WATER TREATMENT SYSTEM (A COMPACT REPRESENTATION)

Timestamp	HMI_Tank_Master_Mode	Tank_Pump_In_Auto	Tank_Pump	Tank_Level
16/06/2017 5:41:08 PM	0	0	0	-8.16274
16/06/2017 5:41:09 PM	1	1	1	-5.092119
16/06/2017 5:41:10 PM	1	1	1	-3.172978
16/06/2017 5:41:11 PM	1	1	1	0.767656

In this section the methodology of converting device logs into event log is described. The method is elaborated with the device logs presented in Table II, a more compact representation of device logs shown in Table I.

A. Event Logs Pre-processing

A device log is a three-tuple $\mathfrak{L}_D = (\mathfrak{T}, \mathfrak{D}_{name}, \mathfrak{D})$, where $\mathfrak{D}_{name} = \{\mathbf{d}_{name\_1}, \mathbf{d}_{name\_2}, \ldots, \mathbf{d}_{name\_M}\}$ is a finite set of system's attribute name, $\mathfrak{T} = \langle \mathrm{t}_1, \mathrm{t}_2, \ldots, \mathrm{t}_N \rangle$ is a sequence of timestamps at which system's attribute value is recorded. While, $\mathfrak{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M\}$ is finite set of sequences, where $\mathbf{d_i}$ represents a sequence of system's attribute value whose length depends on the observation duration such as, $\mathbf{d}_i = \langle d_{i_1}, d_{i_2}, \ldots, d_{i_N} \rangle$ each element $d_i(k)$ represents the ith device status at the kth time instance. Before processing device logs to extract event information, we need to filter these logs.

1) Device Logs Filtering: The device logs contain redundant and unwanted information that doesn't represents any event information. For example, in the QUT water tank dataset shown in Table I, "Tank_Off_SP_Int" and "Tank_On_SP_Int" is unwanted information for event-driven analysis because it represents set-points to maintain the water level. While, redundant status record information such as "Tank_Level" and "Tank_Read_Tank_Level" refer to same record i.e., "level sensor readings". All the unwanted device logs that do not contain any event information are filtered. Redundant information needs to be filtered such that no two "system's attribute vectors" are equal, expressed as follows:

$$\mathfrak{D} = \bigcup_{i=1}^{M} \mathbf{d}\_i \ \mathtt{s.t.}\ \mathbf{d}_i \neq \mathbf{d}_j \ \forall (i,j) \in \mathbb{N} \wedge i \neq j \tag{1}$$

$$\mathfrak{D}_{name} = \bigcup_{i=1}^{M} \mathbf{d}_{name\_i} \ \mathtt{s.t.}\ \mathbf{d}_i \neq \mathbf{d}_j \ \forall (i,j) \in \mathbb{N} \wedge i \neq j \tag{2}$$

2) Events Information Extraction: A hybrid system's attributes are a combination of logical and continuous value vector sets i.e., $\mathfrak{D} = \mathfrak{D}_L \cup \mathfrak{D}_C$, where $\mathfrak{D}_L$ and $\mathfrak{D}_C$ represents the set of logical valued and continuous valued vectors of the system's attribute respectively.

$$\mathfrak{D}_L = \bigcup_{i=1}^{M} \mathbf{d}_i \ \mathtt{s.t.}\ \mathbf{d}_i \in \mathfrak{D} \wedge \mathbf{d}_i \in \{0,1\} \tag{3}$$

$$\mathfrak{D}_C = \bigcup_{i=1}^{M} \mathbf{d}_i \ \mathtt{s.t.}\ \mathbf{d}_i \in \mathfrak{D} \wedge \mathbf{d}_i \in \mathbb{R} \wedge \mathbf{d}_i \notin \{0,1\} \tag{4}$$

For the sake of clarity, the term *continuous value* of attribute is used to refer the real numbered data such as **Tank_Level**. Although the sensor readings are digitised by an analogue to digital converter before being processed by a PLC. However, it shows a continuous trend like increasing or decreasing and any change in value from the previous time record is not an event. We use change point analysis to extract event information.

A change point is defined as the instance in a signal where the statistical parameters (e.g., mean, variance, or slope etc.) of a signal changes abruptly [18]. Based on the experiments on ICS datasets trends, the input and output system attributes either increase, decrease or remain constant as shown in Fig. 2 and 3. Therefore, the statistical parameter 'slope' is well suited to find unknown changes points. This approach is useful for both logical and continuous valued system's attributes. However, in case of continuous variables the detection rate suffers from false positives i.e., change point does not refer to an event as highlighted in Fig. 2. Therefore, we remove any change point in the sequence of events for which the trend (i.e., increasing or decreasing) remains the same.

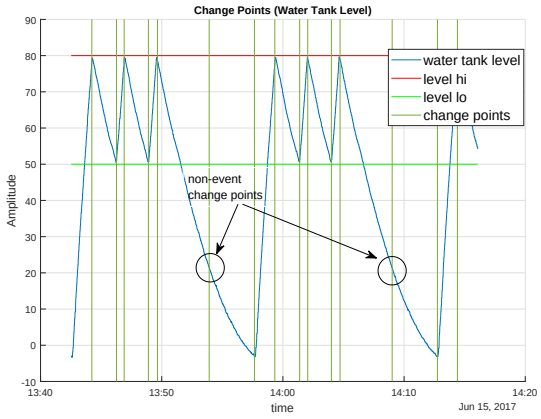

Fig. 2. Change points in water tank level variable

Let $\mathbf{c_i} = \{c_{i_1}, c_{i_2}, \ldots, c_{i_K}\}$ be a sequence of integers that represent change point indices (i.e., occurrence of event) of a signal vector $\mathbf{d_i}$ such that $0 < c_{i_1} < c_{i_2} < \ldots < c_{i_K} < N$. Let $\mathscr{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$ be a finite set of vectors, where $\mathbf{e_i}$ represents an event sequence of the ith system's attribute given by:

$$\mathbf{e}_i = \bigcup_{k=1}^{K} \mathbf{d}_i\left(c_{i_k}\right) \ \mathtt{s.t.}\ \mathrm{sgn}\left(\mathbf{d}_i\left(c_{i_k}\right) - \mathbf{d}_i\left(c_{i_{k-1}}\right)\right) \neq \mathrm{sgn}\left(\mathbf{d}_i\left(c_{i_{k+1}}\right) - \mathbf{d}_i\left(c_{i_k}\right)\right) \tag{5}$$

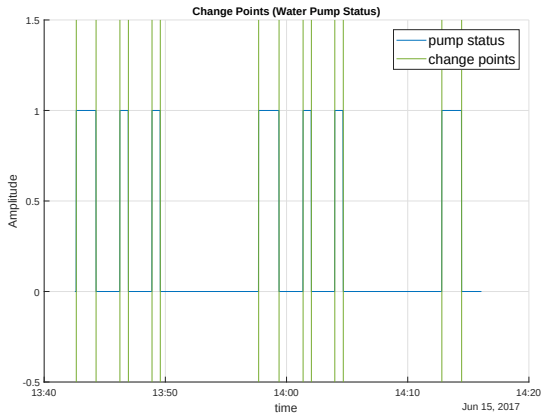

Fig. 3. Change points in water tank level variable

$\mathscr{E}$ represents the status value of system's attributes at the event indices $\mathscr{C}$. These values correspond to a specific event name which has been discussed in Sec. III-A3.

The method of extracting change points is as follows. Let $\mathbf{y} = [y_1, y_2, \ldots, y_N]$ be the signal of interest that changes abruptly at unknown $R-1$ instances, a sequence of points as $\mathbf{k} = (k_0, k_1, \ldots, k_{R-1})$ that divides the signal into R segments. A common approach used to find the change point is to minimise the contrast function J, that measures the deviation in the desired statistical property at each point of the section [18], given by:

$$J(r) = \sum_{r=0}^{R-1} \sum_{i=k_r}^{k_{r+1}-1} \Upsilon\left(y_i; \theta\left(\left[y_{k_r} \ldots y_{k_{r+1}-1}\right]\right)\right) \tag{6}$$

where θ is the characteristic parameter function of signal $\mathbf{y}$ that changes abruptly at the change points. While Υ is the deviation measurement in the characteristic parameter. The deviation in the slope of a signal to find out change points at the ith instance given by [19]:

$$\sum_{i=k_m}^{k_n} \Upsilon\left(y_i; \theta\left(\left[y_{k_m} \ldots y_{k_n}\right]\right)\right) = \left(k_n - k_m\right) \widehat{\sigma}\left(\left[y_{k_m} \ldots y_{k_n}\right]\right) - \frac{\left(\sum_{i=k_m}^{k_n} \left(\left(y_i - \widehat{\mu}\left(\left[y_{k_m} \ldots y_{k_n}\right]\right)\right)\left(i - \mu\left(\left[k_m \ldots k_n\right]\right)\right)\right)\right)^2}{\left(k_n - k_m + 1\right)\sigma\left(\left[k_m k_{m+1} \ldots k_n\right]\right)} \tag{7}$$

where $\widehat{\sigma}$ is empirical variance and $\widehat{\mu}$ is empirical mean. For an unknown number of change points, the contrast function J is minimised by adding a penalty function β to each change point to limit overfitting [20].

$$\mathbf{c} = \arg\min_{r} J(r) + \beta r \tag{8}$$

3) Events Name Derivation: As discussed earlier an event is an instantaneous change in the state of the system. The event information $\mathscr{E}$, $\mathscr{C}$ extracted in the previous section must be named that it is useful for the understanding of discovered model. We attached systems' attribute state with the event information in case of continuous valued system attributes because event information *increasing* or *decreasing* does not provide a full description of the event. For example, every time pump is turned on level starts rising as shown in Fig. 1, thus adding level information can be useful such as *tank_level_lo_&_increasing*. In case of the water tank system example discussed above, **Tank_Level** values during normal system operation at the event instance $\mathbf{e}_i$ be either $80 \pm u_m$, $50 \pm u_m$, *or* $-5 \pm u_m$ which refers to $Tank\_Level\_Hi$, $Tank\_Level\_Lo$, *or* $Tank\_Level\_Empty$, where u_m refers to measurement error. While, in the case of logical valued system attributes such as **Tank_Pump**, it is either *Turned On* or *Turned Off*.

Let $\mathscr{A}_{name}$ be the set of possible elementary event names. For each system's attribute $\mathbf{d}_{name\_i} \in \mathfrak{D}_{name}$ and its specific value $e_{i_k} \in \mathbf{e}_i$, we define an event name derivation function $f_{event}^{\left(d_{name\_i}, e_{i_k}\right)} : \mathfrak{D}_{name} \times \mathscr{E} \rightarrow \mathscr{A}_{name}$ to map $\mathbf{d}_{name\_i}$ and $\mathbf{e}_i$ to elementary event name a_{name}. This function takes the pair $\left(d_{name\_i}, e_i\left(k\right)\right)$ and returns an event name $e_{name\_i_k}$. It is expressed as a set: $\mathfrak{E}_{name} = \{\mathbf{e}_{name\_1}, \mathbf{e}_{name\_2}, \ldots, \mathbf{e}_{name\_M}\}$.

4) Event Log Derivation: Let $\mathfrak{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_M\}$ be set of event sequence timestamps given by:

$$\mathfrak{T}_E = \bigcup_{i=1}^{M} \mathfrak{T}\left(\mathbf{c}_i\right) \ \texttt{s.t.} \ \operatorname{sgn}\left(\mathbf{d}_i\left(c_{i_k}\right) - \mathbf{d}_i\left(c_{i_{k-1}}\right)\right) \neq \operatorname{sgn}\left(\mathbf{d}_i\left(c_{i_{k+1}}\right) - \mathbf{d}_i\left(c_{i_k}\right)\right) \tag{9}$$

Therefore, it can be said that $|\mathfrak{E}_{name}| = |\mathfrak{T}| = I$. Let's assume there is a specific sequence of Case ID $\mathfrak{C}_{id} = \{c_{id\_1}, c_{id\_2}, \ldots, c_{id\_I}\}$ for each pair of timestamps and event. Let $\mathfrak{L}_E$ be a sorted sequence of events against timestamps as shown in Table III. For a sequential process system, Case ID is incremented every time for a specific event that marks the start of each process instance to differentiate process traces. For example, event **HMI_Tank_Master_Mode_Activated** occurred during the start of a new process in event logs as shown in Table III. Case ID is only required for process discovery algorithms, while Petri net model identification requires a sequence of events for model discovery. A comparison of these schemes has been provided in [16]. Finally, a sorted sequence of logs containing timestamps $\mathfrak{T}$, events $\mathfrak{E}_{name}$ and case ID $\mathfrak{C}_{id}$ is expressed as follows:

$$\mathfrak{L}_E = \{(t_i, e_{name_i}, c_{id_i}) \ \forall i \in \mathbb{N} \wedge 1 \leq i \leq I | t_i \leq t_i + 1\} \tag{10}$$

A snippet of event logs derived from device logs (Table. I) of water tank system is provided in Table III

IV. Anomaly Detection

Sec. III provides a formalised method to convert the device logs of a *hybrid system* into event logs that are useful process-based anomaly detection discussed as follows.

TABLE III
A SNIPPET OF EVENT LOGS OF WATER TANK SYSTEM

Time	Event	Case
16/06/2017 5:12:37 PM	HMI_Tank_Deactivate	1
16/06/2017 5:46:41 PM	HMI_Tank_Activate	2
16/06/2017 5:46:42 PM	Pump_Status_ON	2
16/06/2017 5:46:44 PM	Tank_Level_Lo_INC	2
16/06/2017 5:48:25 PM	Pump_Status_OFF	2
16/06/2017 5:48:28 PM	Tank_Level_Hi_DEC	2

A. System Identification

For the Petri net model generation from event logs, one of the existing algorithms is utilised based on the comparison of different discovery methods provided in [16]. The selection of specific algorithm relies on the basis that it should be able to discover generalised behaviour without filtering any infrequent event. Therefore, Inductive Miner with perfect fitness [21] was used for model discovery. The next step is to verify the fitness of the discovered model such that the discovered model can replay all the events in the normal sequence. Therefore, a token replay based conformance checking algorithm [22] is employed. It quantifies the fitness of the model between 0 and 1 such that 1 is for a perfect match.

B. Conformance Analysis

ICS event logs are compared with the discovered model from the event logs of system's normal operation to identify anomalies. However, now the aim is to check that if a specific process instance conform with the expected behaviour in addition to "where it deviates from the expected behaviour?". Therefore, an alignment based conformance checking algorithm [22] has been used to identify anomalies in the event logs.

V. EVALUATION

We evaluate if our new ICS device logs processing method leads to an overall improvement in PN model identification-based anomaly detection method. We compared our approach against the existing device logs processing method in [7]. While, a similar model discovery and conformance checking algorithm was utilised for both event logs (processed by our approach and proposed by Myers *et al.* [7]). Moreover, the proposed method was tested with two different datasets. The first dataset is from water treatment plant which is a six stage system from SUTD, iTrust cyber security research center [13]. While, the other dataset is of an industrial process from a three stage system consisting of a conveyor belt sorting system, a water tank system, and a reactor pressure vessel system. This data set was generated at QUT's SCADA Lab [7].

We developed MATLAB scripts to process the devices logs of both datasets into respective events logs based on our method, outlined in Section III. Both "control (or normal) and attack" files of the dataset were processed using the same script. The processed event logs from the control dataset was used to discover expected behaviour. While, the processed attack dataset was used for conformance analysis i.e., identify anomalies by comparing with expected behaviour. For model discovery and conformance checking we used ProM, an open source framework that contain plugins of model discovery and conformance checking algorithms [24].

Fig. 4. A snapshot of anomaly detection results of first dataset

There were total of 41 attacks in the first dataset [13]. However, five attacks i.e., attack no. 5, 9, 15, and 18 have no impact on system's physical process. Moreover, there is no information available in the dataset to detect attack no. 4 [13]. Thus, there are 35 attacks in first dataset and 21 attacks in the second datasets respectively, that change the physical process of the system. A total 30 out of 35 attacks in the first dataset, and 20 out of 21 attacks in the second dataset were detected. A snapshot of anomalies detected based on PN model discovery and conformance checking using ProM is provided in Fig. 4. Moreover, 5 and 7 false positives were detected in the first and second dataset respectively. Attack no. 13, 14, 24, 29, and 37 in the first dataset remain undetected in first dataset. While, attack no. 21 remained undetected in the second dataset is the process ran longer than expected as shown in Fig. 5. PN model identification methods discover input and output events' relation rather than the timing of the process.

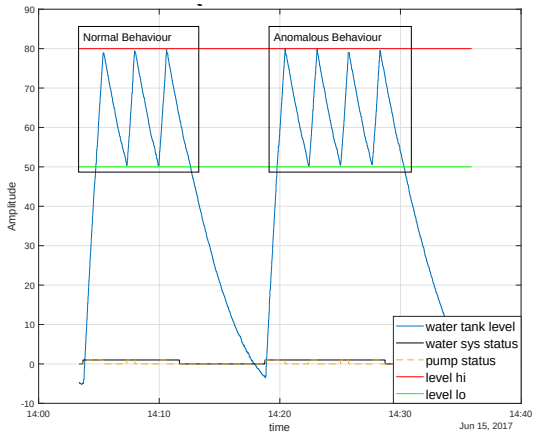

Fig. 5. Attack remained undetected in second dataset [7]

Performance evaluation of anomaly detection is measured using the common criteria 1) True Positive Rate and 2) Precision. Results in comparison with the previous intrusion detection methods for ICS have been summarised in Table IV. It includes a comparison with Myer's device log processing method [7], TABOR [14] and machine learning based IDS method [23]. TABOR is an *automata* model discovery based anomaly detection method [14]. While, machine learning

TABLE IV
EVALUATION RESULTS

	True Positive Rate				Precision			
	This Approach	Myer's [7]	TABOR [14]	CNN [23]	This Approach	Myer's [7]	TABOR [14]	CNN [23]
1st Dataset	0.86	0.33	0.79	0.89	0.86	0.70	0.86	0.95
2nd Dataset	0.95	0.76	–	–	0.71	0.69	–	–

approach is based on convolution neural networks (CNN) for intrusion detection [23]. Our method outperforms previous system modelling based anomaly detection approaches [7], [14] in terms of attack detection. CNN based IDS performs better for attack detection and precision. However, the significance of behavioural modelling-based anomaly detection methods is that they provide necessary information to take swift action on alerts. However, it is not the case for anomaly detection using machine learning algorithms, such as neural networks and n-gram [15].

VI. CONCLUSION & FUTURE WORK

In this paper we have presented a method to convert ICS device logs into event logs that are useful for PN model identification-based anomaly detection. The proposed method in this paper is useful for discovering or rediscovering a PN model from the data generated during system operation instead of relying solely on expert's knowledge. The limitation of manually created PN model is its adaptability to any change in the system configuration. We have presented how to convert continuous valued system attributes for a PN model discovery of a *hybrid control system*. Adding continuous valued system attributes information can significantly improve the anomaly detection results.

In future we plan to expand this research on anomaly detection. The proposed approach relies on the fact that sensor readings are not tampered with. Spoofing attacks in first dataset were not very sophisticated as they do not follow a trend (i.e., increasing or decreasing), just set a sensor reading to a specific value. Thus, these attacks were easily detected by our method. While, the second dataset does not contain any spoofing attacks. Therefore, a promising research area can be the interlacing of process based anomaly detection with network based IDS to improve overall attack detection of sensor spoofing.

REFERENCES

[1] K. Hemsley and R. Fisher, "A History of Cyber Incidents and Threats Involving Industrial Control Systems," in *Critical Infrastructure Protection XII*, pp. 215–242, Springer, Cham, 2018.

[2] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, 2018.

[3] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A Survey of Physics-Based Attack Detection in Cyber-Physical Systems," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–36, 2018.

[4] D. Fauri, D. R. Dos Santos, E. Costante, J. Den Hartog, S. Etalle, and S. Tonetta, "From system specification to anomaly detection (and back)," in *Workshop on Cyber-Physical Systems Security and Privacy*, pp. 13–24, 2017.

[5] L. V. Allen and D. M. Tilbury, "Anomaly detection using model generation for event-based systems without a preexisting formal model," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 42, no. 3, pp. 654–668, 2012.

[6] A. Giua and M. Silva, "Petri nets and Automatic Control: A historical perspective," *Annual Reviews in Control*, vol. 45, pp. 223–239, 2018.

[7] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Computers and Security*, vol. 78, pp. 103–125, 2018.

[8] A. Daneels and W. Salter, "WHAT IS SCADA?," in *International Conference on Accelerator and Large Experimental Physics Control Systems*, (Trieste, Italy), pp. 339–343, 1999.

[9] A. P. Estrada-Vargas, E. López-Mellado, and J. J. Lesage, "A Black-Box Identification Method for Automated Discrete-Event Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1321–1336, 2017.

[10] P. J. Antsaklis and A. Nerode, "Hybrid control systems: An introductory discussion to the special issue," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 457–460, 1998.

[11] B. Vogel-Heuser, C. Diedrich, A. Fay, S. Jeschke, S. Kowalewski, M. Wollschlaeger, and P. Göhner, "Challenges for Software Engineering in Automation," *Journal of Software Engineering and Applications*, vol. 07, no. 05, pp. 440–451, 2014.

[12] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, algorithms, and applications*. Prentice-Hall of Australia Pty. Limited, Sydney Prentice-Hall, 3rd ed., 1996.

[13] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A Dataset to Support Research in the Design of Secure Water Treatment Systems," in *International Conference on Critical Information Infrastructures Security*, pp. 88–99, 2017.

[14] Q. Lin, S. Adepu, S. Verwer, and A. Mathur, "TABOR: A Graphical Model-based Approach for Anomaly Detection in Industrial Control Systems," *ASIA CCS - ACM Asia Conference on computer and communications security*, vol. 12, pp. 525–536, 2018.

[15] S. Etalle, "From Intrusion Detection to Software Design," in *European Symposium on Research in Computer Security*, pp. 1–10, 2017.

[16] M. P. Cabasino, P. Darondeau, M. P. Fanti, and C. Seatzu, "Model Identification and Synthesis of Discrete-Event Systems," in *Contemporary Issues in Systems Science and Engineering*, pp. 343–366, Hoboken, NJ, USA: John Wiley & Sons, Inc., apr 2015.

[17] W. van der Aalst, *Process mining: Data science in action*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2nd ed., 2016.

[18] M. Lavielle, "Using penalized contrasts for the change-point problem," *Signal Processing*, vol. 85, no. 8, pp. 1501–1510, 2005.

[19] "Find abrupt changes in signal," 2016 (Accessed May 10, 2019). https://www.mathworks.com/help/signal/ref/findchangepts.html.

[20] A. Tchorbadjieff and I. Angelov, "Change Point Analysis as a Tool to Detect Abrupt Cosmic Ray Muons Variations," in *Advanced Computing in Industrial Mathematics*, vol. 728, pp. 395–406, Springer International Publishing, 2019.

[21] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach," in *International Conference on Applications and Theory of Petri Nets and Concurrency*, pp. 311–329, 2013.

[22] W. Van der Aalst, A. Adriansyah, and B. Van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, 2012.

[23] M. Kravchik and A. Shabtai, "Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks," in *Workshop on Cyber-Physical Systems Security and PrivaCy*, (New York, USA), pp. 72–83, ACM Press, 2018.

[24] "Process mining research tools and applications," 2016 (Accessed May 10, 2019). http://www.processmining.org/prom/start.