Process Discovery for Industrial Control System Cyber Attack Detection

David Myers(✉), Kenneth Radke, Suriadi Suriadi, and Ernest Foo

Queensland University of Technology, 2 George St, Brisbane, QLD 4000, Australia
d2.myers@qut.edu.au

Abstract. Industrial Control Systems (ICSs) are moving from dedicated communications to Ethernet-based interconnected networks, placing them at risk of cyber attack. ICS networks are typically monitored by an Intrusion Detection System (IDS), however traditional IDSs do not detect attacks which disrupt the control flow of an ICS. ICSs are unique in the repetition and restricted number of tasks that are undertaken. Thus there is the opportunity to use Process Mining, a series of techniques focused on discovering, monitoring and improving business processes, to detect ICS control flow anomalies. In this paper we investigate the suitability of various process mining discovery algorithms for the task of detecting cyber attacks on ICSs by examining logs from control devices. Firstly, we identify the requirements of this unique environment, and then evaluate the appropriateness of several commonly used process discovery algorithms to satisfy these requirements. Secondly, the comparison was performed and validated using ICS logs derived from a case study, containing successful attacks on industrial control systems. Our research shows that the Inductive Miner process discovery method, without the use of noise filtering, is the most suitable for discovering a process model that is effective in detecting cyber-attacks on industrial control systems, both in time spent and accuracy.

Keywords: Industrial Control Systems · Process mining · Anomaly detection

1 Introduction

There have been several cyber-attacks on ICS devices and infrastructure. One notable recent example is the December 23, 2015 attack on three Ukrainian power companies, causing power disruption to 225,000 customers [7]. Such attacks highlight the risk of interconnecting ICS with corporate networks, most notably in terms of the inadequacy of existing IDSs for control system purposes. Corporate networks typically use Intrusion Detection Systems (IDS) to monitor for known cyber-attacks. IDSs come in two types, signature-based IDS (which compare network traffic to signatures of known cyber-attacks), and statistical anomaly based IDS, (which examines network traffic over time for unusual behaviour). However,

S. De Capitani di Vimercati and F. Martinelli (Eds.): SEC 2017, IFIP AICT 502, pp. 61–75, 2017.
DOI: 10.1007/978-3-319-58469-0_5

these IDSs do not detect attacks which exploit or disrupt the *control flow* of an ICS [6], thus, such IDSs may not be sufficient as the goal of exploiting an ICS network is the disruption of ICS system (which is achievable through disruption of the control flow of the systems), rather than exfiltration or infiltration of data as per a typical corporate network attack.

An ICS *control flow* refers to a process, i.e. sequence of events, conducted by an ICS device. These ICS processes are sometimes multiple sequential processes operating concurrently. ICS typically executes a series of steps for a given task, that modify the behaviour of physical devices. Tasks may include creating various fuels in an oil refinery, manufacturing circuit boards, or large scale food manufacturing. The order of events in an ICS process is imperative to the task being conducted, and the deviation from this sequence of events, or control flow, can disrupt the current industrial process. Using the oil refinery example, there may be a sequence of events used to burn off excess gases, such as "turn on flame" → "release gas" → "turn off flame". Changing the order of events to "turn on flame" → "turn off flame" → "release gas" could result in the gas being continually released, potentially damaging equipment. These events may all be valid ICS events, however when the events are conducted in the incorrect sequence there is potential to disrupt the industrial process being conducted.

As per the example above, in order to detect anomalies in ICS system, there is first a need to have knowledge and a model of the expected behaviour of the ICS process. ICS systems are often modelled using state or control flow diagrams, where the actual execution semantics of ICS devices are not captured. This is where techniques from the process mining domain can be of use. Process mining consists of a series of techniques used to *"discover, monitor, and improve real processes by extracting knowledge from event logs"*, and is comprised of process discovery, conformance checking, and process enhancement techniques [14].

Process discovery is the method of deriving a process model describing the control flow of a system from an event log. Using this technique there is the potential to generate process models of an ICS system showing the expected behaviour, by learning from device logs generated by ICS devices such as PLCs. Conformance checking is another technique used to compare a process model or some known business rules to an event log, in order to determine how the events in the log align with the expected behaviours as captured in a process model. In the context of ICS security, we can use conformance checking to identify deviant behaviour that does not match the expected behaviour identified in a generated process model. Thus, by using process discovery and conformance checking, we have a set of tools which can be used to identify deviations, or disruptions in an ICS control flow. In contrast, traditional IDSs which use signatures, or monitor a network for statistical anomalies, do not focus on the overall control flow of a process.

There are numerous process discovery algorithms, each of which have advantages and disadvantages. Several studies have been conducted, comparing these process discovery algorithms [10,17]. However, these comparisons have been conducted in the context of business processes. ICS is a unique environment, and

as such there is a need to have an understanding of which process discovery algorithms are the most suitable for this environment.

Thus, the contributions of this paper are as follows: Firstly, we identify the ICS characteristics and ICS modeling requirements. Secondly, we evaluate process models generated using several widely used process discovery algorithms, namely the α-algorithm [13], the Fuzzy Miner [5], the ILP Miner [16], the Flexible Heuristics Miner (FHM) [18], and the Inductive Miner [11] algorithms, on our ICS process modeling requirements for process discovery. Finally, we validate the evaluation of process models with ICS modeling requirements through process mining's conformance checking activity. This conformance checking activity was conducted using ICS device logs from industry-standard ICS devices, four Siemens S7-1200s, which contain cyber-attacks from an 8-hour attacking vs defending exercise. These process discovery and conformance checking activities show that process discovery algorithms can be used to generate models, and detect cyber-attacks, within an ICS control flow.

2 Background

Industrial control systems (ICS), including supervisory control and data acquisition (SCADA) networks, consist of devices such as programmable logic controllers (PLC) which are controlled by operators through human-machine interfaces (HMI). These devices are used to control industrial facilities and critical infrastructure, such as power generation and distribution systems, and water treatment facilities [4,8].

ICS networks can spread over large geographical regions, and have been traditionally controlled through point-to-point communications such as serial links. However, these networks are increasingly migrating from dedicated communication links to switched and routed corporate networks through the use of specialised gateways, accessed via virtual private networks (VPN) or the Internet [8,9]. While exposure to corporate networks and the Internet allows for easier management of ICS devices over a large geographical distance, it also places these devices at risk of cyber-attack [8,9].

ICS devices generate device logs, which are typically stored on the PLC, or in complex ICS networks, these device logs can be aggregated to a Historian, a device for the storage and archival of device logs. There are several methods ICS devices use to generate device logs, primarily the "Cyclic" method, which records all "tags" or memory values on the PLC at some predefined interval of time, and the "On Change" method, which only records values which have changed in some way. An additional common method of event logging used by ICS devices is to log an event *only when an error occurs*. This method of recording events is not suitable for process mining based analysis, as the error logs do not log the control flow of the ICS system. From observation of the recorded device logs, we found the "on change" device logs did not measure each value of all tags as the value changed, where some tags with high variance (such as measuring pressure, or water level) were recorded in 2–3 s intervals. In this regard, the "Cyclic" method

configured with a 1 s polling interval provides more complete and accurate device logs than the "on change" method. As such, in this paper, we use the Cyclic type of device log as the starting point for process mining based analysis.

Process models, generated by process discovery algorithms, capture the normative behaviour of the system being modeled. As an example of a process model, we have a sequence of events, (A, B, D). In the second iteration of the process, the sequence of events is (A, C, D). This can be represented in a process model using a Petri-net, describing the control flow of the process, as shown in Fig. 1. This method of discovering and modeling the sequence of events can be applied to ICS systems, with the potential to be used to discover a process model, and attacks on the control flow of an ICS process.

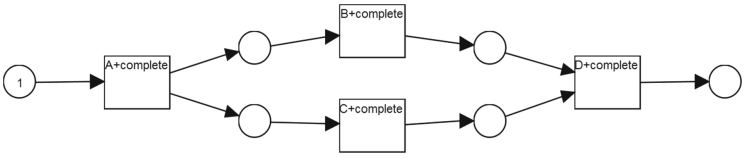

Fig. 1. Example of a process model capturing both sequential and concurrent events.

There are a wide range of process discovery algorithms, which can be identified in several types; such as abstraction-based, heuristic-based, search-based and region-based algorithms [3]. The issue of determining the "best" process mining discovery algorithm for a process discovery activity is a known issue in the process mining domain [12], and several comparisons of algorithms have previously been conducted [10,17]. While process discovery and conformance checking have previously been used in the area of security auditing of business processes to detect anomalous behaviour in various contexts [1–3], they are insufficient in the ICS context. This is because there are a number of differences between business processes, the traditional focus of process mining, and ICS processes. Business processes typically contain events which are executed by humans and recorded by an information system, such as a Workflow Management System [15]. Compared to ICS processes, these business processes are far more flexible and varied than an ICS process, which is a structured and rigid sequence of events executed by a ICS device. In addition to this difference, event logs generated from business processes are typically far more verbose when compared to device logs from ICS devices, such as PLCs. Therefore, there is a need to identify which process mining algorithm(s) are best for discovering and modeling an ICS process.

To determine which process discovery algorithms are most suitable to generate process models from ICS device logs, that can also be used as an input to a conformance checking activity, we have identified several process discovery algorithms for comparison. These discovery algorithms, namely, are the α-algorithm [13], the Integer Linear Programming (ILP) Miner [16], the Flexible Heuristics Miner (FHM) [18], and the Inductive Miner (IM) [11] algorithms.

These algorithms were selected as they appear promising for use in conformance checking to detect cyber-attacks on ICS devices. Before we can assess the algorithms, we need to identify requirements needed to create a suitable model of ICS processes.

3 ICS Modeling Requirements

In the context of ICS networks, there are two main goals for the discovery of an ICS process model. Firstly, to use as an input for a conformance checking activity for the goal of detecting cyber-attacks in an ICS event log, and secondly, to create a process model of an ICS process which is clear and understandable for human interpretation. To generate a process model to suit both of these goals, a process discovery algorithm is used. Process discovery algorithms all have several common features, including the ability to model both sequential and concurrent events, and in addition model repeating event sequences. We have identified several further characteristics for consideration when identifying the most suitable algorithm for ICS process discovery.

Usable Model Representation. For our analysis, we require a usable process model, which contains clear execution semantics, for use in the conformance checking activity. The current state-of-the-art conformance checking activities require the use of Petri-net models. As such, throughout our experiments we use Petri-net models to represent our ICS process and conduct conformance checking activities.

Accuracy. Rozinat et al. [12] outline two evaluation dimensions outline two evaluation dimensions which can be applied to determine the accuracy of a model, "Fitness", and "Precision". The first dimension, fitness, describes how well the event log matches the process model. Fitness is measured numerically, between 0 and 1, where 1 represents a perfect fit. Some process discovery techniques remove "noise" from a process model, commonly to simplify the model. Examples of this are the Fuzzy Miner, and the Inductive Miner. However, this will lower the fitness of the model. For conformance checking purposes, especially for identifying anomalous events, the fitness of the generated process model is imperative. The Precision evaluation dimensions prevents over general models, where the generated models are accurate to the process being modelled.

Simplicty. This requirement reflects the goal of creating a process model of an ICS process which is clear and understandable for human interpretation. In most ICS networks, there does not exist a process model of either individual process, or an overall model. The generated process model must be represented in a way which is understandable, avoiding overcomplicated "spaghetti" models. In the case of anomaly detection, when a process model is visually complex, it becomes difficult for a human to quickly visually validate the anomaly. Rozinat et al. [12], describe two evaluation dimensions which can apply to this simplicity requirement, "Generalisation" and "Structure". The Generalisation evaluation dimension, conversely to "Precision", prevents over precise models, where over precise

models may show each path in the process and may contain duplicate events, which may be easier to understand but less fitting. The Structure dimension refers to the visual structure of the process model. Highly fitting process models may be visually complicated, and hard to read and interpret. Modeling repeating event sequences is a common feature of all process discovery algorithms, and can be modeled either directly in the model, or expressed through the use of silent (τ)-transitions (suitable when there is a choice in the model). Silent transitions are not necessary to model repeating event sequences in a process model, however make the model less complicated. These evaluation dimensions can have an effect on the accuracy of the process model, where more generalisation can reduce the overall fitness of the model.

4 Experiment

In this section, we outline our experimental methodology to evaluate process models by generated by discovery algorithms, and compare to our identified ICS characteristics and ICS modeling requirements. Firstly, we outline the experimental setup, describing the ICS devices, logging method, and dataset used throughout our experiment. Secondly, we describe our process discovery, and validation using conformance checking activities.

4.1 Experimental Setup

The experimental setup consists of four distinct scale industrial control systems, each of these controlled by industry-standard PLCs, Siemens S7-1200's. These four scale systems include a bi-directional conveyor belt system, a water pump system, and a "reactor" system, all connected to a master power meter. These four systems are controlled through the use of a HMI, which acts as a historian, collecting device logs from each of the PLCs. These four devices together with the HMI made a "Process Control Network", connecting to a multi-level corporate network through a gateway, representing a typical ICS network. The device logs recorded throughout our experiment are "Cyclic" device logs, which were configured to record with a 1 s interval, and stored in comma-separated values (CSV) format. This "Cyclic" device log method is where the value of all "tags", or addresses in a PLCs memory, are recorded every specified interval, in our configuration, every 1 s. An example of a typical, unprocessed ICS device log is shown below, in Table 1.

The scale industrial systems form one complete industrial process. This industrial process starts with the bi-directional conveyor belt system, which consists of two loops. Two sensors; one which detects the presence of a puck, and one which detects the colour of a puck, control the direction of a "paddle". As an example of the "Cyclic" device logging method, a snippet of the device log from the conveyor belt is shown in Table 1. Once every second all "tags", or memory addresses on the PLC logged. When a puck is detected, the "VarValue" of the tag "Conv_Read_Conv_Color_PE", shown in Table 1, will change from 0 to 1

Table 1. Example snippet of a typical ICS device log, obtained from the conveyor belt system.

VarName	TimeString	VarValue	Validity	Time_ms
Conv_Read_Conv_Color_PE	16/07/2015 9:31	0	1	42201396613
Conv_Read_Conv_HMI_Direction	16/07/2015 9:31	0	1	42201396613
Conv_Read_Conv_Present_PE	16/07/2015 9:31	0	1	42201396613
Conv_Read_Solenoid_Left_Direction	16/07/2015 9:31	0	1	42201396613
Conv_Read_Solenoid_Right_Direction	16/07/2015 9:31	0	1	42201396613

or −1 depending on the colour of the puck. Once a puck is detected and the colour is determined, the paddle will direct the puck onto one of the two conveyor belt loops depending on the colour.

Once the conveyor belt system has completed, the water tank system begins to operate. The water tank system consists of an upper reservoir and a lower reservoir full of water. The lower reservoir is connected to a water pump, which transfers water to the upper reservoir, which uses gravity to feed water back to the lower reservoir. The upper water reservoir contains a water sensor, which turns off the pump when the specified water level is met. This process repeats until a specified time length has elapsed.

The final stage of the industrial process is the "reactor", a pressure vessel system. The reactor is a pipe system, which is connected to an air compressor, and a solenoid with pressure sensors that measures the pressure level in the pipe system, in pounds per square inch (PSI). Once a specified pressure level has been reached, the solenoid activates and opens its valve, releasing air and lowering the pressure to a lower level. Once the lower level has been reached, the solenoid closes and the "reactor" begins to build pressure once again. This process repeats for a designated period of time, at which the remaining pressure is released and the air compressor switches off.

The device logs were generated by the PLCs throughout the operation of an 8 h exercise, in which two teams: a defending team of four people protecting the process control and corporate networks from an attacking team of six attackers with industrial experience. The attackers were tasked with disrupting the process, through exploiting the "Process Control Network", and causing the ICS devices to deviate from the programmed, expected behaviour of that device.

One example of the attacking team successfully disrupting the industrial process is the changing of the direction of the bi-directional conveyor belt. The conveyor belt sorts coloured "pucks" placed on the belt, sending each colour around a different loop on the conveyor. The attacking team successfully compromised the HMI, and changed the sorting direction of these coloured pucks. This results in the pucks being directed around the wrong direction, disrupting the expected industrial process. Throughout this exercise the attacking team were successfully able to disrupt the industrial process running on all three systems, the conveyor belt system, the water tank system, and the "reactor"

system. The device logs from each ICS system were collected by the HMI on the "Process Control Network", which acted as the historian. These collected devices logs contained the cyber-attacks that were conducted by the attacking team throughout the duration of the 8 h exercise. These device logs were then aggregated and pre-processed, creating our dataset of event logs, and preparing them for process mining activities.

The dataset consisted 25 cases, with 71 event classes and a total of 1855 events. One "case" represents a complete execution of the ICS process. The PLCs, HMI, and device logging methods were programmed and configured through the use of WinCC and STEP 7, as part of the Siemens TIA Portal V11 SP2. The experiments were not conducted on production ICS networks, as attacks conducted upon these scale systems interrupt the control flow of the ICS process. These scale systems and industrial process used throughout or experiments, while not production systems, use industry-standard ICS devices, four Siemens S7-1200 PLCs.

4.2 Experimental Methodology

Using our dataset of pre-processed event logs, generated from device logs aggregated throughout the 8 h exercise, we discover 5 process models using the widely used algorithms outlined in Sect. 2. These include the α-algorithm, ILP Miner, Flexible Heuristics Miner, Inductive Miner configured with default fitness, and Inductive Miner configured with perfect fitness. The default and perfect fitness for the Inductive Miner are configuration parameters when generating a process model. The use of both Inductive Miner with default fitness and the Inductive Miner with perfect fitness to generate a process model is to compare the results of removing noise and low-varying events from a process model, and the impact this has upon conformance checking for anomaly detection. In addition to the models generated by process discovery algorithms, we use a manually-created Petri-net model of the ICS process. While the manual creation of a process model can capture an accurate picture of an ICS process, this requires specialist knowledge of the ICS systems, and process, being modeled. Using process discovery algorithms, a process model of a complicated ICS system can be created without this in-depth knowledge requirement.

To discover a process model using the outlined algorithms from our dataset, we extracted the first 10 cases, or first 10 complete process executions of the ICS process; a representative sample of the complete ICS process where the cases were *known to contain no cyber-attacks*. It is assumed while generating models of the ICS process, the event logs used in the generation contain no attack data. Using the Disco[1] tool, we converted the pre-processed event log from CSV format to XES format, for use with the Process Mining Toolkit (ProM)[2] version 6.5.1. Using the Petri-net models generated using the ProM tool, and our manually created Petri-net model, we conducted a total of 6 conformance checks, using ProM with the

[1] http://fluxicon.com/disco/.
[2] http://www.promtools.org/doku.php.

"Replay a Log on Petri Net for Conformance Analysis" plug-in, comparing each discovered process model against our full dataset. Both the process discovery and conformance checking activities were conducted on a Dell Optiplex 9020 desktop computer, with an i7-4770 CPU, 16 GB RAM, and 256 GB SSD.

5 Results and Analysis

We have laid out in Sect. 3 the requirements for generating process models, firstly for use in a conformance checking activity with the goal of identifying anomalies, such as cyber attacks, in an ICS event log, and secondly for human interpretation. The process models generated by the selected process mining algorithms, outlined in Sect. 2 are evaluated by these requirements, and the results of this evaluation are described under each requirement below, and summarised in Table 2.

Table 2. Comparison of models generated by selected process discovery algorithms with identified requirements of ICS process discovery.

Algorithm	Model	Accuracy	Simplicity
α-algorithm [13]	✓	✓	✗
ILP Miner [16]	✓	✓	✗
FHM [18]	✓/✗	✗	✓
IM (Default) [11]	✓	✗	✓
IM (Perfect) [11]	✓	✓	✓

All models were either generated in or can be converted to Petri-net form, making all models usable in a conformance checking activity. To test the "Fitness" of the generated process models, we conduct a conformance check on the representative sample used to generate each model. The process models generated by the ILP Miner, α-algorithm, and Inductive Miner with perfect fitness, all fit the representative example of the ICS process event log which was taken to generate the process models, and are precise. Both models generated by the Flexible Heuristics Miner with a move-log fitness of 0.27, and the Inductive Miner, configured with default fitness, with a move-log fitness of 0.95, do not fit this sample. This is due to both algorithms filtering infrequent events. The process model generated for use in conformance checking for anomaly detection must be as close to, or perfectly fitting, to the ICS process as possible. Any non fitting events during a later conformance check for anomaly detection are treated as "anomalous", resulting in "false positives" as the model is inaccurate. This can be observed in Table 3. From this "fitness" metric, we can determine that the process models generated by the Flexible Heuristics Miner and Inductive Miner with default fitness are not suitable for use in a conformance checking activity to detect anomalous events, such as cyber attacks (Fig. 2).

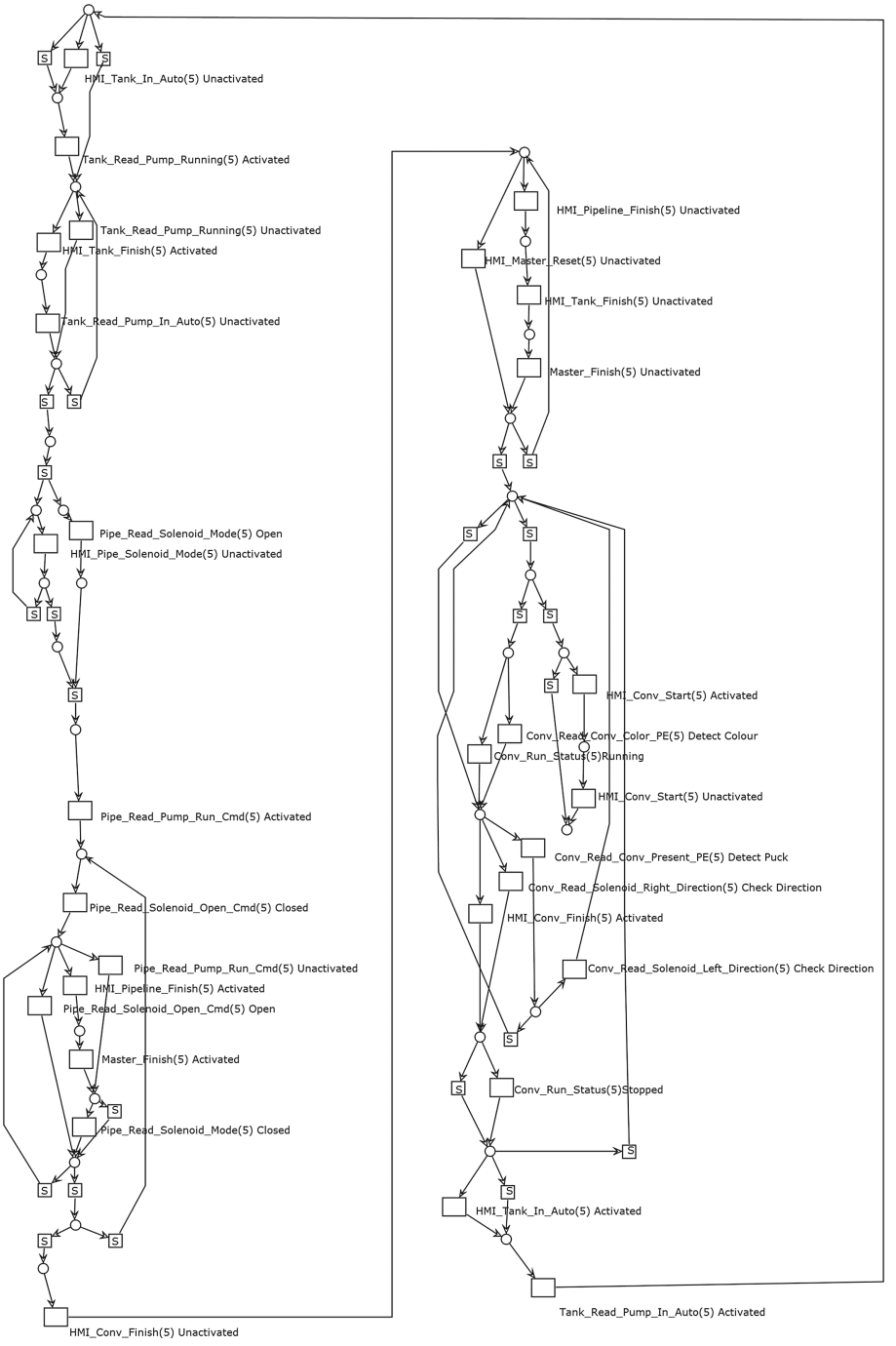

Fig. 2. Process model generated using Inductive Miner (perfect fitness).

We have identified two evaluation dimensions for the Simplicity requirement, described in Sect. 3, "Generalisation" and "Structure". Both the generalisation and structure dimensions can be evaluated through observation of the generated process models. The process models generated by the Heuristics Miner, and Inductive Miner are both structured and are at a suitable level of generalisation, showing the control flow of the ICS process without impacting on the fitness of the process model. However, the models generated by the α-algorithm and ILP Miner, although fitting, have significantly more arcs within the model, with 47 places connected by 122 arcs, and 26 places connected by 179 arcs respectively. This significant difference in arcs is shown in Fig. 3. This is compared to Inductive Miner with 30 places connected by 88 arcs for default fitness, and 34 places connected by 110 arcs for perfect fitness, resulting in the ILP and α-algorithm models being less structured and visually harder to interpret. The process models generated by the Flexible Heuristic Miner and Inductive Miner with both default fitness and perfect fitness settings model repeating event sequences through the use of silent transitions. These models with silent transitions simplify the model, resulting in the models being easier to interpret.

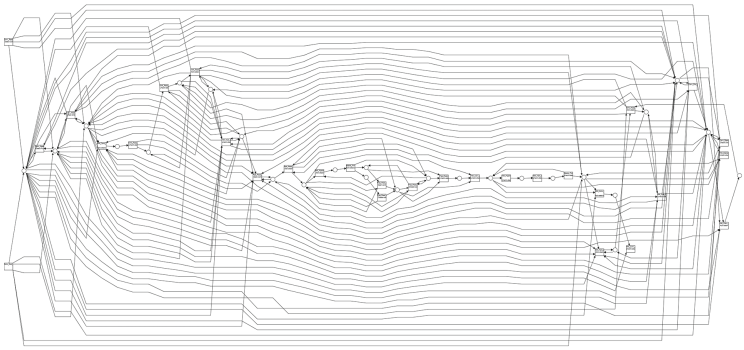

Fig. 3. Process model for illustrative purposes using the ILP Miner to show the complexity of discovered model, where there is less generalisation and structure, with and more arcs in model.

Using the process models generated from several widely used discovery algorithms, we conduct several conformance checking activities on our pre-processed ICS event log. Our dataset contained 25 cases, five of these cases contained cyber-attacks which were conducted on the experimental setup throughout the duration of the 8-hour exercise, outlined in Sect. 4.1. The results of these conformance checking activities are shown below, in Table 3, including number detected anomalous cases, Trace Fitness, and calculation time in milliseconds. The "Trace Fitness" value represents how well the event log fits the Petri-net models generated by the discovery algorithms.

The conformance checking activities were successfully able to identify the 5 anomalous cases on models generated by the α-algorithm, ILP Miner, Inductive

Miner with perfect fitness, and our existing, manually-created model, as shown in Table 3. These anomalous cases contain the cyber-attacks conducted by the attacking team throughout the 8 h exercise, showing where the event log deviates from the expected behaviour of the ICS, outlined in the generated process models. An example of this deviation can be shown with the water tank system. During one attack case, the attackers exploited the HMI and changed the operation of the water tank from "automatic" to "manual". This allows an attacker to manually control the level at which the water level sensor activates, potentially allowing the attackers to overflow one of the water reservoirs. When this event is recorded in the device log, and "replayed" over the process models during the conformance checking activity, it appears as a deviation where the event did not "fit" the process model representing the control flow of the ICS process.

Table 3. Results of conformance checking activity on discovered process models.

Algorithm	Cases	Trace Fitness	Time(ms)
α-algorithm [13]	5	0.93	25.60
ILP Miner [16]	5	0.93	2.08
FHM [18]	25	0.31	54.24
IM [11] (Perfect)	5	0.93	1.88
IM [11] (Default)	24	0.89	1.48
Manual Model	5	0.93	28.68

Both models created by the Flexible Heuristics Miner and Inductive Miner with default fitness settings failed to accurately model the ICS process, resulting in the conformance check returning both 25 and 24 anomalous cases, respectively. This was caused by both Flexible Heuristics Miner and Inductive Miner reducing "noise".

The Flexible Heuristics Miner, and Inductive Miner with default fitness, both employ methods of filtering noise and low-frequency events. While the removal of noise is a common and useful practice in process mining activities, it is not recommended in anomaly detection for identifying cyber-attacks, as the models must accurately represent the behaviour of the ICS. Removing low-frequency events or "noise" from the model can result in false positives, as shown with Flexible Heuristics Miner and Inductive Miner (with default fitness). These results suggest that the Inductive Miner algorithm, configured with perfect fitness, is the most suitable process discovery algorithm for both the creation of a process model of an ICS process for human interpretation, and for use in a conformance checking activity with the goal of identifying anomalies, such as cyber attacks, in a ICS event log. One limitation to current process discovery algorithms is the concept of a "case" in process mining. Most ICS device logs do not have an indication of "case", and require a pre-processing stage to prepare the logs for process mining activities. An area of future work is the development of a

process discovery algorithm which can discover an ICS process model without the requirement for a "case".

In addition to detected cases, we found considerable differences in the time required for conducing the conformance checks. The ProM tool reported the calculation time in milliseconds(ms) taken for each full conformance check. The conformance check on the model generated by the Flexible Heuristics Miner was conducted in the largest amount of time, a total of 54.24 ms. This was followed by the existing, manually-created model in 28.68 ms, and the α-algorithm in 25.60 ms. The conformance checks using the models generated by the Inductive Miner (with perfect fitness), the Inductive Miner (with default fitness), and ILP Miner, were significantly faster. The Inductive Miner with perfect fitness completing in 1.88ms, with default fitness in 1.48 ms, and the ILP miner in 2.08 ms. The conformance check upon the manually-created model the returned same anomalous cases as process models generated by the α-algorithm, ILP Miner, and Inductive Miner configured with perfect fitness, however took more time to complete, in 28.68 ms. This indicates the process models generated by process discovery algorithms can accurately capture an ICS process.

Existing IDS used on ICS networks rely on signatures of known attacks in signature-based IDS, or long-term analysis of network traffic in statistical anomaly based IDS. For a signature of an cyber-attack to exist, there must be prior knowledge of the cyber-attack. Our process mining based process discovery and conformance checking methods used in this paper have successfully detected cyber-attacks conducted on industry standard ICS devices, the Siemens S7-1200, using process models generated by widely-used process discovery algorithms. This shows process mining can be used in supplement to traditional IDS devices to detect previously-unknown cyber-attacks.

6 Conclusion

The conformance checking activity with the α-algorithm, ILP Miner, Inductive Miner with perfect fitness all successfully identifying the five anomalous cases containing cyber attacks in the ICS event log. Experimental results show that the Inductive Miner algorithm, configured with perfect fitness, was the most suitable algorithm for use in detecting cyber attacks.

The process mining based methods presented in this paper, unlike IDS systems, are a form of offline analysis, which takes place on device logs after a process has been completed. An area of future work is to modify this method to operate in real or near-real time to improve the detection speed of ICS anomalies. Some attacks, such as Man-in-the-Middle (MITM) attacks used in the Stuxnet attack, may not be detected due to insufficient logging. One method of detecting these attacks could be to correlate device log data and low level sensor data for use in process mining based intrusion detection. Larger datasets, and assessment of the scalability of this method on larger datasets is required.

References

1. Van der Aalst, W.M.P., de Medeiros, A.K.A.: Process mining and security: detecting anomalous process executions and checking process conformance. Electron. Notes Theor. Comput. Sci. **121**, 3–21 (2005)
2. Accorsi, R., Stocker, T.: On the exploitation of process mining for security audits: the conformance checking case. In: SAC, pp. 1709–1716. ACM (2012)
3. Accorsi, R., Stocker, T., Müller, G.: On the exploitation of process mining for security audits: the process discovery case. In: SAC, pp. 1462–1468. ACM (2013)
4. Daneels, A., Salter, W.: What is SCADA. In: Bulfone, D., Daneels, A. (eds.) International Conference on Accelerator and Large Experimental Physics Control Systems, pp. 339–343. ELETTRA, October 1999
5. Günther, C.W., Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007). doi:10.1007/978-3-540-75183-0_24
6. Hadžiosmanović, D., Bolzoni, D., Hartel, P.H.: A log mining approach for process monitoring in SCADA. Int. J. Inf. Secur. **11**(4), 231–251 (2012)
7. ICS-CERT. Alert (IR-ALERT-H-16-056-01) cyber-attack against ukrainian critical infrastructure. https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01, Accessed 18 Apr 2016
8. Igure, V.M., Laughter, S.A., Williams, R.D.: Security issues in SCADA networks. Comput. Secur. **25**(7), 498–506 (2006)
9. Knijff, R.V.D.: Control systems/SCADA forensics, what's the difference? Digital Invest. **11**(3), 160–174 (2014). Special Issue, Embedded Forensics
10. Leemans, M., van der Aalst, W.M.P.: Discovery of frequent episodes in event logs. In: Ceravolo, P., Russo, B., Accorsi, R. (eds.) SIMPDA 2014. LNBIP, vol. 237, pp. 1–31. Springer, Cham (2015). doi:10.1007/978-3-319-27243-6_1
11. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Lohmann, N., Song, M., Wohed, P. (eds.) BPM 2013. LNBIP, vol. 171, pp. 66–78. Springer, Cham (2014). doi:10.1007/978-3-319-06257-0_6
12. Rozinat, A., de Medeiros, A.K.A., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The need for a process mining evaluation framework in research and practice. In: Hofstede, A., Benatallah, B., Paik, H.-Y. (eds.) BPM 2007. LNCS, vol. 4928, pp. 84–89. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78238-4_10
13. Van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
14. van der Aalst, W.M.P., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28108-2_19
15. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., de Medeiros, A.K.A., Song, M., Verbeek, H.M.W.E.: Business process mining: an industrial application. Inf. Syst. **32**(5), 713–732 (2007)

16. van der Werf, J.M.E.M., van Dongen, B.F., Hurkens, C.A.J., Serebrenik, A.: Process discovery using integer linear programming. In: Hee, K.M., Valk, R. (eds.) PETRI NETS 2008. LNCS, vol. 5062, pp. 368–387. Springer, Heidelberg (2008). doi:10.1007/978-3-540-68746-7_24
17. Weerdt, J.D., Backer, M.D., Vanthienen, J., Baesens, B.: A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. Inf. Syst. **37**(7), 654–676 (2012)
18. Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible heuristics miner (FHM). In: CIDM, pp. 310–317. IEEE (2011)