The International Journal of Advanced Manufacturing Technology (2022) 118:2467–2479
https://doi.org/10.1007/s00170-021-08001-6

ORIGINAL ARTICLE

Automated detection-in-depth in industrial control systems

Zahra Jadidi[1] · Ernest Foo[2] · Mukhtar Hussain[1] · Colin Fidge[1]

Received: 1 March 2021 / Accepted: 2 September 2021 / Published online: 30 September 2021

Abstract

Legacy industrial control systems (ICSs) are not designed to be exposed to the Internet and linking them to corporate networks has introduced a large number of cyber security vulnerabilities. Due to the distributed nature of ICS devices, a detection-in-depth strategy is required to simultaneously monitor the behaviour of multiple sources of ICS data. While a detection-in-depth method leads to detecting attacks, like flooding attacks in earlier phases before the attacker can reach the end target, most research papers have focused on anomaly detection based on a single source of ICS data. Here, we present a detection-in-depth method for an ICS network. The new method is called automated flooding attack detection (AFAD) which consists of three stages: data acquisition, pre-processing, and a flooding anomaly detector. Data acquisition includes data collection from different sources like programmable logic controller (PLC) logs and network traffic. We then generate NetFlow data to provide light-weight anomaly detection in ICS networks. NetFlow-based analysis has been used as a scalable method for anomaly detection in high-speed networks. It only analyses packet headers, and it is an efficient method for detecting flooding attacks like denial of service attacks, and its performance is not affected by encrypted data. However, it has not been sufficiently studied in industrial control systems. Besides NetFlow data, ICS device logs are a rich source of information that can be used to detect abnormal behaviour. Both NetFlow traffic and log data are processed in our pre-processing stage. The third stage of AFAD is anomaly detection which consists of two parallel machine learning analysis methods, which respectively analyse the behaviours of device logs and NetFlow records. Due to the lack of enough labelled training datasets in most environments, an unsupervised predictor and an unsupervised clustering method are respectively used in the anomaly detection stage. We validated our approach using traffic captured in a factory automation dataset, Modbus dataset, and SWAT dataset. These datasets contain physical and network level normal and abnormal data. The performance of AFAD was compared with single-layer anomaly detection and with other studies. Results showed the high precision of AFAD in detecting flooding attacks.

Keywords Industrial control systems · Anomaly detection · NetFlow-based analysis · Log analysis · Histogram clustering · Prediction

1 Introduction

The Purdue model of a typical industrial control system (ICS) architecture [1, 2] provides six levels for interconnected information technology (IT) and operational technology (OT) devices: level 0 (field devices like sensors and actuators), level 1 (local controller like programmable logic controllers (PLCs)), level 2 (supervisory control and data acquisition (SCADA) components and distributed control systems (DCSs)), level 3 (control centre and processing LAN), and levels 4 and 5 (enterprise zone). SCADA systems in level 2 are deployed for high-level monitoring and management of critical infrastructures. A DCS is a network of control devices which are parts of one or more industrial processes. PLCs are early control systems interacting with physical devices, and they provide local management of processes being run through feedback control devices such as sensors and actuators.

Advancements in the field of networks have enabled engineers to remotely access ICS devices by connecting them to corporate networks. However, this exposes ICSs to the Internet and makes them vulnerable to cyber-attacks.

✉ Zahra Jadidi
zahra.jadidi@qut.edu.au

[1] Cyber Security Cooperative Research Centre, Queensland University of Technology (QUT), Queensland, Australia

[2] Griffith University, Brisbane, Australia

Studies show that the number of documented attacks on ICS infrastructures has increased dramatically in recent years [2]. This has raised concerns for the security of ICSs. Advanced multi-step ICS attacks may compromise several devices discussed above, and hence, their malicious activities may be reflected in different ICS data sources like network traffic (captured from Ethernet-based components), and device logs.

Simultaneous monitoring of all data sources in ICS networks helps security analysts to detect attacks in earlier stages before an irreparable security breach happens. However, the majority of anomaly detection methods proposed during the past few years are based on local analysis and their solutions are not applicable to large-scale and multi-level ICS networks with different types of data sources. Multi-layer analysis has been deployed in a number of papers to detect ICS attacks [40, 43–45]. However, they utilised multiple methods to improve the accuracy of anomaly detection in a single type of data source. In this paper, we propose a multi-layer detection or detection-in-depth method for ICS networks to analyse data sources received from ICS layers and detect attacks in earlier stages.

There are some methods proposed for the detection of advanced persistent threats (APTs) which are multi-step attacks in IT and ICS networks [47–49]. Detection based on attack signatures is a method proposed for APT attacks. APT attacks have dynamic behaviours, and they follow diverse attacking techniques and tactics to reach their goal. Therefore, detection based on attack signatures is not efficient for zero-day attacks [47]. Two other methods proposed for an ICS environment are correlation analysis and causality analysis of loges [48, 49]. While these methods have been deployed in different papers to detect APT attacks, authors assumed that they already have a pre-existing intrusion detection strategy in each layer, and hence, they try to correlate security and non-security logs to visualise attackers' paths and predict future actions. However, in this paper, we aim to improve the efficiency of detection-in-depth in ICS networks.

Lightweight anomaly detection in the early levels of an ICS network helps to detect a multi-step attack quicker before it can reach physical devices. Despite packet-based analysis, NetFlow-based methods process only packet headers and they provide light-weight analysis. The efficiency of NetFlow-based anomaly detection is illustrated in IT networks [6–8, 46]. However, it has not been investigated in an ICS environment. NetFlow-based analysis can detect flooding attacks which affect packet headers quicker and earlier than payload-based IDSs. NetFlow-based anomaly detection in ICS networks is the second knowledge gap investigated in our paper. In this paper, we present a detection-in-depth method that exploits both NetFlow traffic and physical device logs to detect attacks. Our physical log analysis is built on the analysis of ICS device logs that record the state of physical devices such as sensors and actuators [3–5]. Therefore, our solution can separately monitor these important types of ICS data and increase the chance of detecting anomalous behaviour.

While supervised learning methods in anomaly detection may provide greater precision of anomaly detection [22, 27], labelling data for these methods is very expensive and time-consuming and, consequently, they are not practical for high-speed networks in the real world. Our study used two types of unsupervised methods to analyse sensor logs and NetFlow traffic. In our multi-layer anomaly detection, the first layer is an unsupervised clustering method used to identify malicious NetFlow records. The second layer is based on behavioural pattern analysis of physical devices. The ARIMA/GARCH predictor method is employed to create a model of normal behaviour and predict future values [11]. Any deviation in the actual value from the predicted one is identified as an anomaly.

The novelty of our paper is presenting a detection-in-depth method which is able to monitor distributed ICS data sources including NetFlow data. To the authors' knowledge, this is the first work investigating the efficiency of NetFlow-based anomaly detection in an industrial control environment containing various data sources and legacy systems. As NetFlow-based analysis is only based on header data, it reduces the volume of data, and encryption of payloads does not affect its outcomes. Moreover, NetFlow data can be easily generated and collected by network devices such as routers and switches. Using NetFlow analysis and log analysis, our multi-layer approach can perform significantly better than anomaly detection based solely on either network packets or device logs. To demonstrate this, our method was evaluated using three publicly available datasets, which includes a factory automation dataset [12], a Modbus dataset [13], and Singapore University of Technology and Design (SUTD) dataset [14]. Compared with a single layer anomaly detection, our detection-in-depth method is capable of detecting advanced attacks in earlier levels of ICS architecture, and prevent them from launching their next attacks.

2 Background and related work

The main difference between IT and ICS networks is the latter's integration into OT physical devices. Due to this integration, ICSs have introduced new types of vulnerabilities, and attacks on these weaknesses may cause anomalous behaviour. There are various data sources in ICS networks that can be monitored for anomaly detection [15, 16]. Different methods have been published for anomaly

detection in a single source of ICS data [17, 22, 27]. For example, several machine learning (ML)–based methods have been separately proposed for anomaly detection in physical-layer log features (Q learning [18], Dyna-Q [19], Linear SVM [23], BDT [20] and neural networks [24]). Anomaly detection based on network-traffic features is also studied in a number of papers (artificial neural networks [21]). Anomaly detection based on physical-layer features uses the predictable behaviour of industrial control systems to detect anomalous behaviour [22]. This behavioural-based anomaly detection has been used by Dong et al. [4] who proposed an intrusion detection method based on mapping the periodic traffic characteristics into a hash digest using a discrete cosine transform and singular value decomposition. Markman et al. present another notable work [25], in which periodicity of communications between PLCs and HMI/Engineering workstations can be modelled as deterministic finite automata. While there are multiple levels of ICS devices, the solutions provided by these previous papers do not scale to large-scale ICS networks.

For multi-layer ICS networks, a detection-in-depth strategy is required. While a network-based approach can be deployed for anomaly detection in ICS networks [8, 9, 12], the inherited problem with network traffic based anomaly detection algorithms is that they cannot identify anomalous behaviours of compromised workstations at the supervisory control layer. For example, an attacker may get access to an operator's credentials and use them to send commands to disrupt the physical process. To address such a challenge, researchers have proposed physical process-based anomaly detection algorithms [12, 26, 27]. This approach can identify attacks from compromised workstations. However, such algorithms are not efficient in detecting DoS and distributed denial of service (DDoS) attacks affecting the volume of network traffic in topper levels of ICS networks. A DDoS or DoS attack on an ICS can be efficiently detected at the network layer before it affects the physical process. Therefore, we observe that a detection-in-depth method that takes into account both network traffic data and physical device logs can be efficient and effective. This is the first gap addressed in our paper.

Network traffic analysis is divided into payload-based analysis and packet header-based analysis (NetFlow-based analysis). Payload-based anomaly detection in ICS environment is a well-studied area. However, NetFlow-based anomaly detection in ICS is a research gap that is considered in our paper. NetFlow-based analysis can be used to detect cyber-attacks which affect the volume of network traffic. This method is capable of detecting attacks like port scanning, DNS Poisoning, DoS, and DDoS attacks [6, 7, 29, 30]. The lack of payload analysis makes the NetFlow-based methods scalable, fast and cost-effective for anomaly detection [41]. Thus, NetFlow-based analysis can detect flooding attacks in the earliest stages before they affect the entire network. This method can be used as a complementary method to payload-based anomaly detection methods.

NetFlow is a Cisco proprietary protocol, which can be enabled on router devices, to provide NetFlow records. A NetFlow record is defined as a group of packets with some common characteristics which pass a monitoring point in a specific time interval [6, 8, 9]. Compared with payload-based methods, a NetFlow-based analysis method significantly reduces the volume of traffic to be processed. For example, in an IT network at the University of Twente, the ratio between traffic exported by NetFlow and packets on the network was 0.1 [10]. NetFlow-based analysis has been employed in different papers to detect anomalies in IT networks [6–9, 29, 41]. However, our work is the first investigating the efficiency of NetFlow-based anomaly detection in an ICS environment.

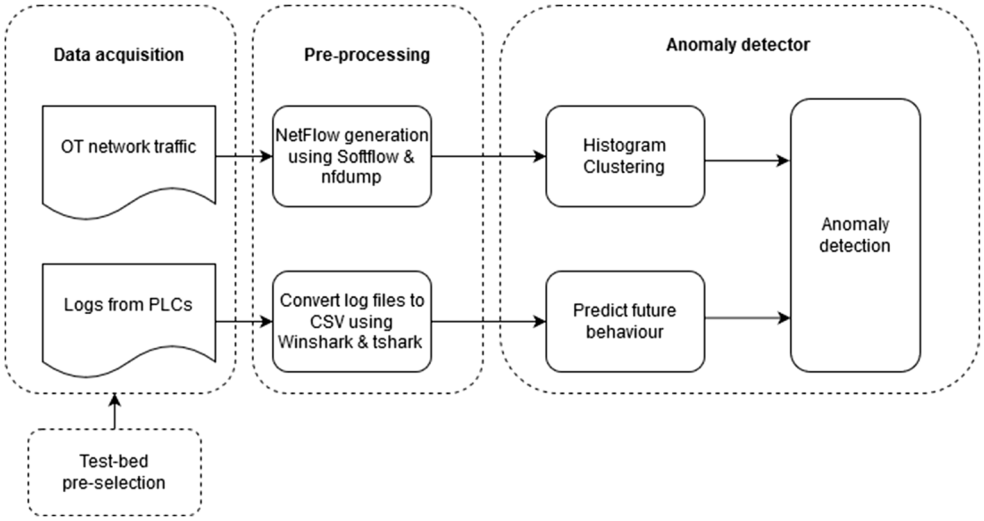
3 Detection-in-depth anomaly detection approach

The architecture of our automated flooding attack detection (AFAD) method is shown in Fig. 1. Our approach is composed of three consecutive stages: data acquisition, pre-processing, and anomaly detection. In the data acquisition stage, ICS device logs and network packets are captured. In the pre-processing stage of AFAD, network packets are processed by NetFlow simulators (Softflow and nfdump) to generate NetFlow records which can be used by the detection phase. ICS device logs are also processed to remove any redundant information and generate CSV files readable by ML-based predictor. The final detection phase is two-layer analysis. The first layer detects abnormal NetFlow records using histogram clustering [8, 28]. The second layer predicts the baseline of a PLC's normal behaviour using the ARIMA/GARCH method. An anomaly alert is generated if either the clustering or prediction methods find an anomaly in the input data.

The novelty of this architecture is use of a detection-in-depth method in ICS networks by the integration of NetFlow analysis with log-based anomaly detection. NetFlow-based analysis provides light-weight anomaly detection as it only processes packet headers. Two-layer anomaly detection in AFAD can be applied to levels 1 to 3 of OT networks of the Purdue model. As shown in Fig. 2, log predictor analyses logs from level 1, and NetFlow-based anomaly detection can monitor network traffic in levels 2 and 3.

To provide an overview of the proposed AFAD approach, the pseudo-code of this approach is provided in Table 1 to

Fig. 1 The AFAD architecture



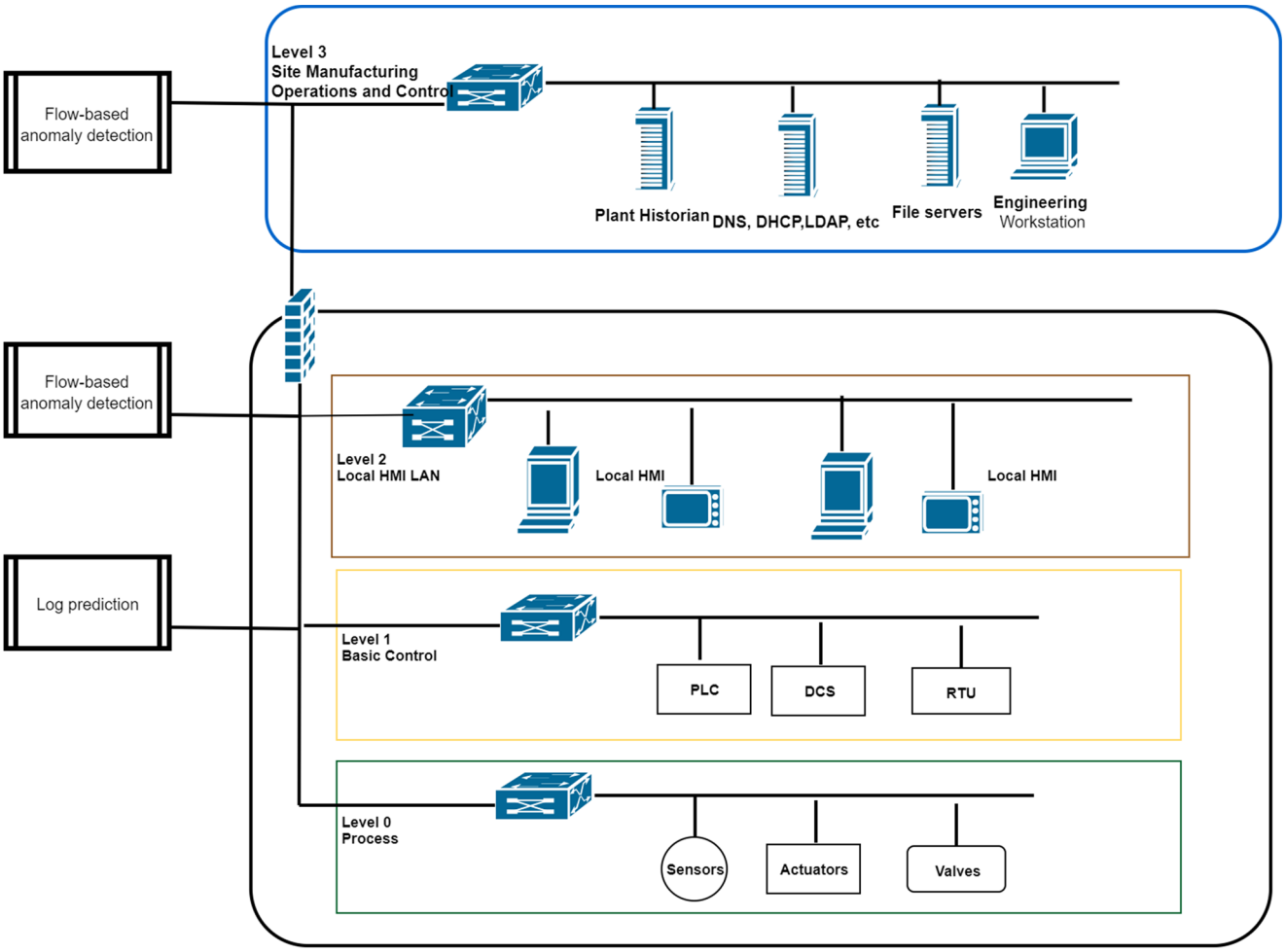


Fig. 2 Secure Purdue model for industrial networks

Table 1 AFAD pseudo-code

Pseudo-Code: AFAD algorithm
Input: Input data from industrial control system levels (Network traffic (Pcap) and logs)
Output: Anomaly detection in ICS levels
// Pre-processing
While not at end of network packets **do**
read network Pcap files;
foreach specified time interval in Pcap files **do**
convert each group of packets with common features passing a monitoring point into a NetFlow sample
end
end
convert log files to a CSV format
//Anomaly detection
foreach NetFlow sample **do**
turn each sample into a cluster using HCA clustering
Merge the pairs of clusters with the smallest distance
Continue until two clusters (normal and anomaly) are left
end
foreach row in a log **do**
predict the future values of logs using ARIMA/GARCH method
calculate the distance between predicted values and actual logs
detect anomalous logs in which the distance is more than a specified threshold
end
if HCA or ARIMA/GARCH detect an anomaly **then**
AFDA output shows an anomaly
end

summarise the different steps involved in this detection-in-depth strategy. The anomaly detection deployed in AFAD will be discussed in Section 3.3.

3.1 Data acquisition

As discussed earlier, our method provides anomaly detection at level 1 to 3 of the Purdue model for effective attack detection. We used three different datasets to evaluate our method. These datasets contain network traffic packet captures (Pcaps) and physical device logs, one from normal operations and another from abnormal or attacked operations. Details of these datasets are as follows:

- The first dataset (factory automation dataset) was generated from a three-stage, laboratory-based factory automation process consisting of a conveyor belt sorting system, a water tank system, and a pressure vessel system. The process is automated using SIEMENS PLCs. Thus, the communication protocol is S7comm [12]. All the attacks on the system were flooding attacks from an attacker machine on the same network.
- The second dataset (Modbus dataset) was generated using a simulated liquid pump controlled by a PLC. A simulated Modbus RTU device was used to measure liquid temperature thresholds and determine motor speed. The communication protocol used for this dataset was Modbus [13]. All the attacks on the system were flooding attacks (namely Modbus Query flooding, TCP SYN flooding and ping flooding) from an attacker machine on the same network.
- The third dataset (SWAT dataset) is of a small scale industrial water treatment process, which is a six stage system. The dataset was generated by the iTrust Cybersecurity Research Center [14]. The communication protocol used for the automation was Modbus. The attacks on the system were spoofing and man-in-the-middle.

Our detection-in-depth method is initially explained using the factory automation dataset, and the results are then compared with the other datasets and other studies. Due to the lack of data from level 3 in the ICS datasets available to us, NetFlow records were only generated using level 2

traffic. However, the NetFlow based analysis provided in this paper is applicable to both levels 2 and 3.

3.2 Pre-processing

This section describes the pre-processing phase which generates input data for two layers of anomaly detection in our AFAD method. Firstly, NetFlow traffic generation required for histogram clustering is explained. Secondly, pre-processing of physical device logs is discussed.

3.2.1 NetFlow generation

NetFlow-based analysis operates on NetFlow records that have been received in fixed-length time intervals (5 minutes in our experiments). Pre-selection in AFAD is used to make data selections based on the destination of data to be processed. Pre-selection aims to filter out unhelpful data and send the reminder to the detection phase to find anomalies. Therefore, only data that belongs to the pre-selected tuples will be analysed.

In any TCP/IP communication between two hosts, A and B, the TCP session will contain two NetFlow records, one for the traffic going from Host A to Host B, and a second for the traffic going from Host B to Host A. NetFlow creates a NetFlow record for each direction of communication within a TCP traffic session, capturing a standard set of information based on the particular version of NetFlow that is used. For example, using NetFlow Version 5, NetFlow records contain the following information about traffic sessions between hosts: Source IP Address, Destination IP Address, IP Protocol, Source port (for UDP or TCP flows, 0 for other protocols), Destination port (for UDP or TCP, or 0 for other protocols), IP Type-Of-Service flags, TCP Flags, Total Packets in Flow, Total Bytes in Flow, Packets Per Second (PPS), Duration (milliseconds), etc [8, 9, 31–33]. The Softflowd and nfsen tools were used in our work to convert packet datasets to NetFlow version 5 records, as shown in Fig. 3.

Total packets in a flow (packet No.) and total bytes in a flow (flow size) are two important parameters used for NetFlow-based flooding attack detection. Figure 4 uses

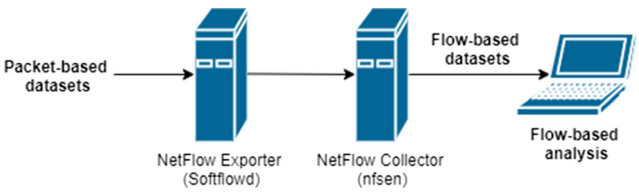


Fig. 3 Flow generation process

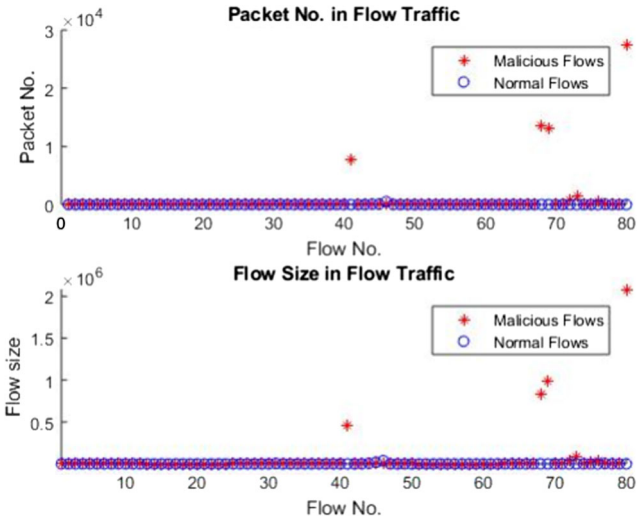


Fig. 4 Comparison between normal and malicious NetFlow records

these parameters to compare normal and malicious flows in the factory automation dataset. Large flows in this figure belong to flooding attacks.

3.2.2 Device logs

ICS device logs are recorded at regular intervals during system operation and stored in a Historian archive. It contains the status of physical devices (sensors and actuators) with timestamps. A snippet of device logs from the factory automation dataset is shown in Table 2. Device logs are normally provided in Pcap formats. Our pre-processing phase used Wireshark (https://www.wireshark.org/) and tshark (https://tshark.dev) software to convert these Pcap files to CSV files required for the anomaly detection phase.

Table 2 A snippet of ICS device logs from the factory automation dataset

Var Name	Time String	Var Value
Tank_Usage_Level(5)	16/06/2017 17:41	108.1627
HMI_Tank_Master_Mode(5)	16/06/2017 17:41	0
Tank_Level(5)	16/06/2017 17:41	−8.162743
Tank_Off_SP_Int(5)	16/06/2017 17:41	80
Tank_On_SP_Int(5)	16/06/2017 17:41	50
Tank_Read_Pump_In_Auto(5)	16/06/2017 17:41	0
Tank_Read_Pump_In_Manual(5)	16/06/2017 17:41	0
Tank_Read_Pump_Running(5)	16/06/2017 17:41	0
Tank_Read_Tank_Level(5)	16/06/2017 17:41	−8.162743
Tank_Stopped(5)	16/06/2017 17:41	−1
Tank_Usage_Level(5)	16/06/2017 17:41	108.1627

3.3 Anomaly detection

This section explains the unsupervised machine learning approaches used in the anomaly detection phase of the AFAD method. Clustering histograms and time series forecasting analyse NetFlow traffic and device logs respectively. The output from AFAD indicates an anomaly if either of these layers detect an anomaly.

3.3.1 Clustering histograms

It has been shown that NetFlow-based analysis can provide a high detection rate in flooding attacks [8]. However, sometimes fluctuations in network traffic can increase the number of false positives and false negatives [8]. Hofstede et al. [8] solved this problem by analysing NetFlow data with histograms that show both the size of a flow and payload size distribution. Inspired by their paper, a similar method was used in our AFAD system to remove the network fluctuation created by TCP re-transmission and control information. Our approach analyses NetFlow data enhanced with histograms that describe packet payload distributions. Using such a histogram, AFAD can discriminate TCP control information and re-transmission from other traffic, and hence, the fluctuations cannot impact NetFlow-based analysis. To extract the per connection histogram from network traffic, our histograms were manually extracted for every observed NetFlow record. This manual histogram extraction was done just for this experiment, and it can be automated in future.

Different researchers have used histograms for packet-based intrusion detection [8, 28, 34], and have investigated how to map network traffic to histograms, cluster the histograms and classify abnormal traffic patterns using these clusters. AFAD applies histogram clustering to NetFlow traffic in ICS networks. A pivotal distance metric can be used for clustering histogram similarity with respect to their bins. Figure 5 shows three payload size histograms we created from the NetFlow-based dataset generated from packet traffic in the factory automation dataset. It is shown that the distance between histograms A and B to be smaller than the distance between histograms A and C.

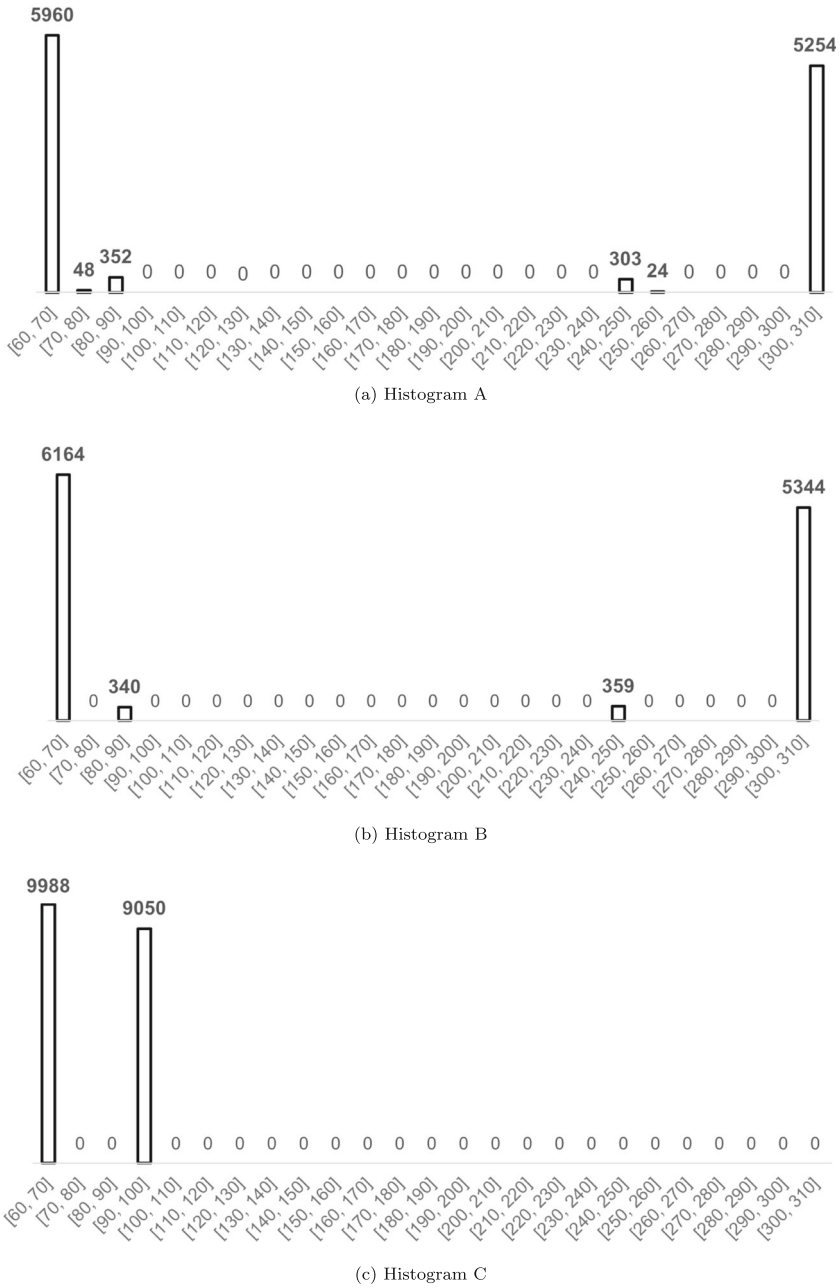
In our experiments, we used Hierarchical Cluster Analysis (HCA) in which inter-cluster distances are used to create clusters in a hierarchical fashion [8, 35]. In HCA, the number of clusters should not be set in advance. This makes HCA different from other clustering methods. A linkage method determines which histograms can be linked clusters and uses inter-cluster distance calculations. A single-linkage method and Minimum Difference of Pair Assignments (MDPA) distance metric were used in our study. Single-linkage calculates the inter-cluster distance using the two least dissimilar histograms in two clusters. Figure 6 shows the output of the HCA clustering we performed on the factory automation dataset.

This figure shows inter-distance or dissimilarity of each histogram to other histograms. HCA builds a tree diagram which puts most similar histograms in groups that are close together. Histograms which join together sooner are more similar. HCA in AFAD clusters puts histograms into two main groups, normal and anomaly, shown as cluster A and B. The performance of this histogram clustering is discussed in Section 4.

3.3.2 Time series forecasting for future PLC logs

The non-linear time series ARIMA/GARCH model was used in our study to predict the expected future values of PLC logs from related variables. This model combines the linear time series ARIMA model with the non-linear GARCH model [11, 36, 37, 42]. The ARIMA model can be employed for time series forecasting, when there is correlation between values during different time frames. The GARCH model was proposed to overcome the shortcomings of the ARIMA model. The GARCH model provides a better description of time series features, including more pre-fluctuation information, and it allows the conditional variance to depend on the previous values. The ARIMA/GARCH hybrid model is a combination of the ARIMA model and the GARCH model, and it could significantly improve the prediction accuracy.

Typically, ICS device logs follow a predictable model of expected behaviour. Traditional methods used manual processes to discover this behavioural model, and significant domain knowledge of the ICS system and industrial process was required for this process. An ARIMA/GARCH model, which is an unsupervised machine learning method, was adopted in our approach to learn and predict the expected behaviour of ICS logs automatically. Therefore, it can detect deviations from the model of expected behaviour, and sends alerts of anomalous behaviours. PLC logs can be stored on the device or through an HMI which communicates with PLCs based on commands and logs [12]. Wireshark, which is a network traffic analyser, was installed on the HMI in the factory automation dataset [12] to capture the Siemens S7-1200 PLCs' logs. However, these logs are not usable for a machine learning algorithm and pre-processing activities such as converting Pcap files to CSV files and normalisation of data are required. Normalization prepares data for a ML method by transforming all input features to the same scale. This improves the training stability of the ML method [38].



(a) Histogram A

(b) Histogram B

(c) Histogram C

Fig. 5 Payload size histograms in ICS flooding attack traffic

4 Experimental results

To evaluate our AFAD method, the factory automation, Modbus and SWAT datasets, described in Section 3.1, were used. The factory automation dataset included two data sources, network Pcap files and log files. However, the Modbus and SWAT datasets only contain one data source which is network Pcap files and PLC logs respectively. Factory automation and Modbus network Pcap files were imported into a NetFlow generator (Fig. 3) to generate NetFlow-based datasets. The numbers of NetFlow records extracted from these datasets are shown in Table 3. To evaluate the unsupervised histogram clustering method in the AFAD approach, 80% of each dataset was randomly selected for a training dataset used for building the model (estimating its parameters) and the remaining 20% was used for testing the performance of the model. The ratio of benign to malicious flows in each NetFlow-based dataset is based on the NetFlow records provided by its original dataset. The same ratio was used for both the training and testing sets.

Device logs were the second data source analysed by ARIMA/GARCH model in our AFAD method. The pattern of normal and malicious logs in the Siemens Master PLC in the factory automation dataset is compared in Fig. 7. Flooding attacks in the factory automation dataset aim to change or disrupt the running industrial process. As is

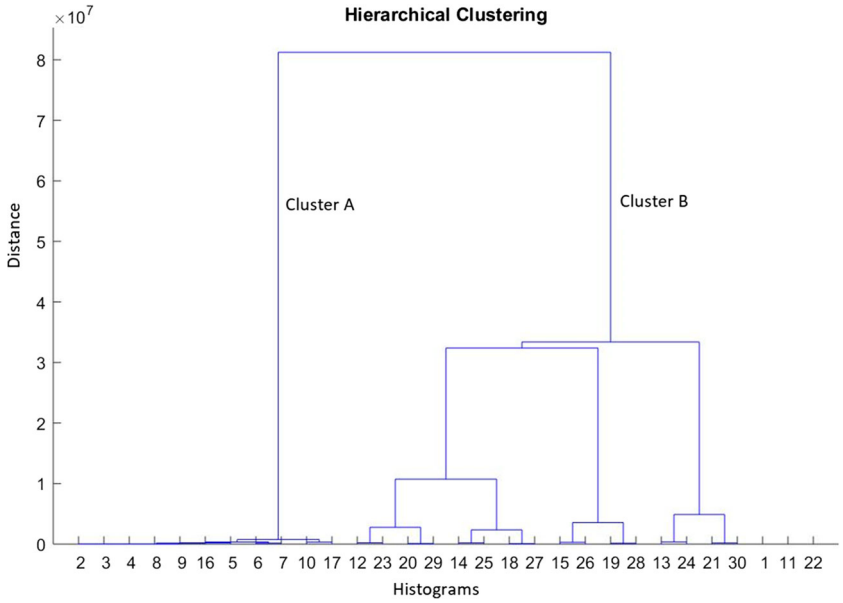


Fig. 6 Hierarchical clustering

Table 3 NetFlow datasets

NetFlow-based dataset	Normal flows	Malicious flows	Total flows
Factory automation dataset	4972	5496	10468
Modbus dataset	9000	9000	18000

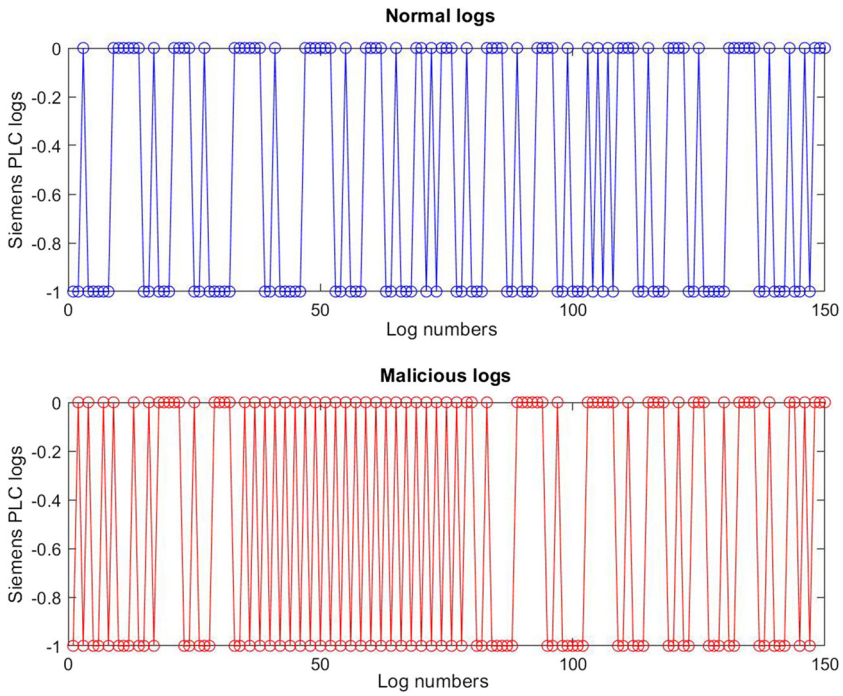


Fig. 7 Comparison between the behavioural patterns of normal and malicious PLC logs

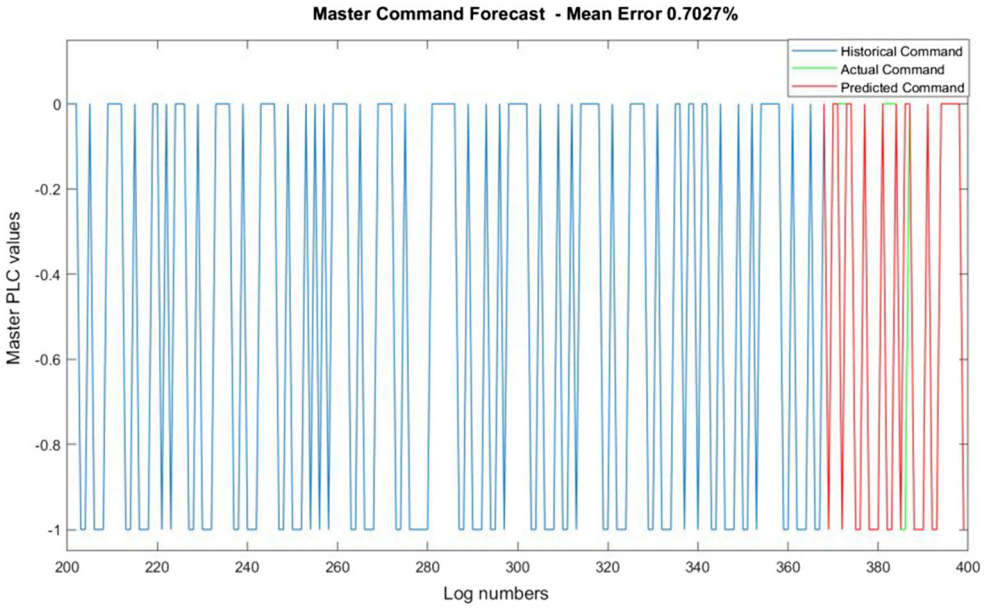


Fig. 8 Predicted values

shown in this figure, these attacks have affected the pattern of Siemens PLC's changes compared with the normal traffic.

The data exchanged between the HMI and PLCs in the factory automation dataset was monitored and used for forecasting future behaviour. Figure 8 shows the output of the ARIMA/GARCH predictor in the factory automation dataset. This figure compares the actual behaviour and the predicted behaviour as well as computing the forecast error. It can be seen that the data predicted by the ARIMA-GARCH model is consistent with the actual log data, and the mean absolute error between the two is only 0.7%. This shows that ARIMA/GARCH can accurately predict future log values. This layer of analysis is able to detect the attacks which are not detected by Netflow-based anomaly detection.

The evaluation metrics of our AFAD system are shown in Table 4. For measuring the performance of AFAD, two metrics are used, Precision and Recall. These evaluation metrics are defined as equations 1 and 2:

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

where True Positive (TP) and True Negative (TN) show correct detection of malicious and normal traffic respectively. False Positive (FP) is the wrong detection of normal traffic, and False Negative (FN) is an anomaly detected as normal. Recall shows our method's ability to detect anomalies. On the other hand, Precision shows the percentage of correct alarms generated by our system. High Precision means a low false alarm rate.

The performance of the AFAD system with the factory automation dataset is compared with single-source anomaly detectors and the results show our approach outperforms

Table 4 Validation results for AFAD

Dataset	Analysis method	TP	FP	FN	Precision	Recall
Factory automation dataset	AFAD system	0.86	0.04	0.1	0.96	0.91
Factory automation dataset	A single flow-based anomaly detector	0.74	0.12	0.3	0.86	0.72
Factory automation dataset	A single log-based anomaly detector	0.76	0.11	0.26	0.88	0.75
Modbus dataset	AFAD system	0.69	0.14	0.13	0.83	0.84
SWAT dataset	AFAD system	0.79	0.07	0.1	0.92	0.91
Modbus dataset	K-Nearest Neighbors [13]	...	...	...	0.75	...
SWAT dataset (LIT101)	SVM [23]	...	...	...	...	0.94
SWAT dataset	MLP [39]	...	...	...	0.96	0.69

single-source analysis. The precision is improved by almost 10%. The first single-source detector in Table 4 corresponds to the HCA clustering layer of AFAD which is used for flow-based anomaly detection. HCA in AFAD clusters histograms into two main groups, normal and anomaly. An ARIMA/GARCH model, adopted in the second layer of AFAD, is the second single-source anomaly detector in this table. The ARIMA/GARCH model is used to learn and predict the expected behaviour of ICS logs automatically. Therefore, it can detect deviations from the pattern of the expected behaviour, and sends alerts of anomalous behaviours. As shown in Table 4, high precision in each layer could improve the overall Precision in the two-layer AFAD. AFAD provides high Precision and Recall, and it can detect attacks earlier in ICS networks. This method is able to detect anomalies. However, it cannot identify the type of attacks occurring in an ICS network.

The Modbus dataset was also used for the evaluation of the NetFlow-based anomaly detector, and the performance of the AFAD was compared with another study [13] in which packet-based anomaly detection was performed using K-Nearest Neighbors (KNN). The results show that the precision of NetFlow-based analysis in AFAD is 8% greater than KNN-based packet analysis.

The SWAT dataset was the third dataset used to evaluate the performance of the AFAD in attacks like Man-in-the-Middle (MitM) attacks. These attacks are not detectable by NetFlow-based analysis as they do not affect packet headers. The AFAD architecture is designed to provide two-layer anomaly detection. NetFlow-based analysis should be able to detect flooding attacks in levels 2 and 3 of the Purdue model, before the attacker can reach PLCs. However, if an attack is not detectable by NetFlow-based analysis, the second layer of detection will be log-based detection. The SWAT dataset was used to show how AFAD can detect attacks which affect PLC behaviour without changing NetFlow traffic. The SWAT dataset provides both normal and attack logs. The LIT101 sensor, a level transmitter sensor, was affected by a MitM attack in this dataset. Figure 9 shows the difference between the LIT101 log behaviour in normal and MitM traffic.

The performance of our AFAD process in MitM detection is shown in Table 4. The precision of AFAD in detecting MitM attacks, based on only device logs, is 92% and recall is 91%. These results were compared with other ML-based anomaly detection methods evaluated by SWAT dataset [23, 39]. Our unsupervised AFAD could provide high recall, 91%, comparable with a supervised SVM's Recall in LIT101. In addition, this recall is 20% more than the MLP neural network in Shayga et al.'s experiment [39].

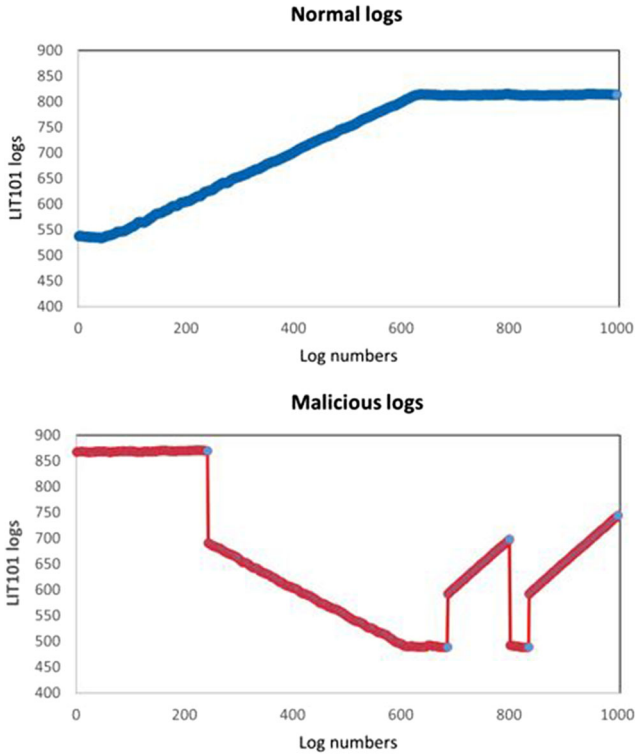


Fig. 9 SWAT dataset

Integration of NetFlow-based analysis and log analysis in the AFAD approach can provide a detection-in-depth strategy. Our method can complement existing signature-based IDSs in industrial networks. The results in this paper showed the high performance of AFAD in detecting both flooding attacks and attacks affecting packet payloads like MitM attacks.

The advantage of our method is the capability of NetFlow-based anomaly detection in detecting flooding attacks earlier in ICS networks, and hence, we can have a quicker response to the attack before the Physical Level of the ICS network is affected.

Two challenging problems in ICS networks are the lack of labelled datasets and the multi-level nature of ICS networks. In this paper, we developed a multi-layer anomaly detection method based on NetFlow analysis and log analysis for ICS networks. Our proposed method can monitor ICS levels simultaneously and detect attacks in the early layers of the ICS network. Based on the authors' knowledge, NetFlow-based anomaly detection in ICSs has not been investigated in any study. The second layer of our proposed AFAD method is responsible for detecting attacks which are not identified by NetFlow analysis. The lack of labelled datasets in ICS environments is also addressed in this study by deploying two unsupervised methods. These unsupervised methods showed high performance in NetFlow and log data analysis, shown in Table 4.

5 Conclusion

Contemporary industrial control networks have different types of sensors and network logs containing information about normal and malicious activities. This study presented a multi-layer analysis technique to improve the visibility of ICS devices and help anomaly detection in an industrial environment. The NetFlow-based analysis employed in our technique helped to detect flooding attacks in top levels of ICS networks. Then, log analysis was used to monitor physical layer logs. Three packet-based datasets were used to generate NetFlow records and ICS logs. Our layered anomaly detection method consisting of unsupervised histogram clustering and an unsupervised predictor could detect flooding attacks and deviation from normal behaviours in PLC logs. The use of histogram clustering allowed us to overcome the problem of fluctuations in normal NetFlow traffic. This study showed that a combination of NetFlow-based analysis and log prediction can be used in real-world industrial networks to detect attacks affecting multiple devices. The experiments showed the promising results of our AFAD system compared with single-source analysis.

This is the first work investigating the efficiency of NetFlow-based anomaly detection in an industrial control environment containing various data sources and legacy systems. While supervised learning methods in anomaly detection may provide greater precision of anomaly detection, labelling data for these methods is very expensive and time-consuming and, therefore, they are not practical for high-speed networks in the real world. Our study used two types of unsupervised methods to analyse sensor data and NetFlow traffic to produce a more practical process.

Acknowledgements The authors acknowledge the support of the Commonwealth of Australia and Cybersecurity Research Centre Limited.

Funding The research was supported by Commonwealth of Australia and Cybersecurity Research Centre.

References

1. SANS Institute: Reading Room - Industrial Control Systems / SCADA, https://www.sans.org/readingroom/whitepapers/ICS/secure-architecture-industrial-control-systems-36327
2. Team UICER (2016) Recommended practice: improving industrial control systems cyber security with defense-in-depth strategies. Retrieved from: www.ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf
3. Hu Y, Yang A, Li H, Sun L, Sun Y (2018) A survey of intrusion detection on industrial control systems. International Journal of Distributed Sensor Networks, pp 1–14
4. Dong R, Wu D, Zhang Q, Zhang T (2018) Traffic characteristic map-based intrusion detection model for industrial Internet. International Journal of Network Security, pp 359–370
5. Hussain M, Foo E, Suriadi S (2019) An improved industrial control system device logs processing method for process-based anomaly detection. In: Frontiers of Information Technology, IEEE
6. Jadidi Z, Muthukkumarasamy V, Sithirasenan E, Singh K (2016) Intelligent sampling using an optimized neural network. Journal of Networks 11(1):16
7. Jadidi Z, Muthukkumarasamy V, Sithirasenan E, Singh K (2015) Flow-based anomaly detection using semisupervised learning. In: 2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS) (pp. 1–5). IEEE
8. Hofstede R, Jonker M, Sperotto A, Pras A (2017) Flow-based web application brute-force attack and compromise detection. Journal of network and systems management 25(4):735–758
9. Hofstede R, Pras A, Sperotto A, Rodosek GD (2018) Flow-based compromise detection: lessons learned. IEEE security & privacy 16(1):82–89
10. Sperotto A, Pras A (2011) Flow-based intrusion detection. In: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops (pp. 958-963). IEEE
11. Babu CN, Reddy BE (2014) A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. Applied Soft Computing 23:27–38
12. Myers D, Suriadi S, Radke K, Foo E (2018) Anomaly detection for industrial control systems using process mining. Computers & Security 78:103–125
13. Frazão I, Abreu PH, Cruz T, Araújo H, Simões P (2018) Denial of service attacks: detecting the frailties of machine learning algorithms in the classification process. In: International Conference on Critical Information Infrastructures Security (pp. 230–235). Springer, Cham
14. Goh J, Adepu S, Junejo KN, Mathur AA (2017) Dataset to support research in the design of secure water treatment systems. In: International Conference on Critical Information Infrastructures Security. pp 88–99
15. Colelli R, Panzieri S, Pascucci F (2018) Exploiting system model for securing CPS: the anomaly based IDS perspective. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA) (Vol. 1, pp. 1171–1174). IEEE
16. Zhang F, Kodituwakku HADE, Hines JW, Coble J (2019) Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Transactions on Industrial Informatics 15(7):4362–4369
17. Wang Q, Chen H, Li Y, Vucetic B (2019) Recent advances in machine learning-based anomaly detection for industrial control networks. In: 2019 1st International Conference on Industrial Artificial Intelligence (IAI) (pp. 1-6). IEEE
18. Xiao L, Li Y, Liu G, Li Q, Zhuang W (2015) Spoofing detection with reinforcement learning in wireless networks, in 015 IEEE Global Communications Conference (GLOBECOM). IEEE, pp 1–5
19. Xiao L, Li Y, Han G, Liu G, Zhuang W (2016) Phy-layer spoofing detection with reinforcement learning in wireless networks. IEEE Transactions on Vehicular Technology 65(12):10 037–10 047
20. Choi W, Joo K, Jo HJ, Park MC, Lee DH (2018) Voltageids: low-level communication characteristics for automotive intrusion detection system. IEEE Transactions on Information Forensics and Security 13(8):2114–2129
21. Sestito GS, Turcato AC, Dias AL, Rocha MS, da Silva MM, Ferrari P, Brandao D (2018) A method for anomalies detection in real-time ethernet data traffic applied to profinet. IEEE Transactions on Industrial Informatics 14(5):2171–2180

22. Hadeli H, Schierholz R, Braendle M, Tuduce C (2009) Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In: IEEE Conference on Emerging Technologies & Factory Automation. IEEE, 2009, pp 1–8
23. Ahmed CM, Zhou J, Mathur AP (2018) Noise matters: using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS. In: Proceedings of the 34th Annual Computer Security Applications Conference ACM, pp 566–581
24. Yang J, Zhou C, Yang S, Xu H, Hu B (2018) Anomaly detection based on zone partition for security protection of industrial cyber-physical systems. IEEE Trans Ind Electron 65(5):4257–4267
25. Markman C, Wool A, Cardenas A (2017) A new burst-DFA model for SCADA anomaly detection. In: Workshop on Cyber-Physical Systems Security and PrivaCy - CPS '17
26. Kreimel P, Tavolato P (2019) Neural net-based anomaly detection system in substation networks. In: 6th International Symposium for ICS & SCADA Cyber Security Research, 2019(6), pp 41–48
27. Liu J, Guo J, Orlik P, Shibata M, Nakahara D, Mii S, Takac M (2018) Anomaly detection in manufacturing systems using structured neural networks. In: 13th World Congress on Intelligent Control and Automation (WCICA)
28. Kind A, Stoecklin MP, Dimitropoulos X (2009) Histogram-based traffic anomaly detection. IEEE Trans. Netw. Serv. Manag. 6(2):110–121
29. Karasaridis A, Meier-Hellstern K, Hoein D (2006) Detection of DNS anomalies using flow data analysis, Global Telecommunications Conference, 2006. GLOBECOM'06
30. Li B, Springer J, Bebis G, Gunes MH (2013) A survey of network flow applications. Journal of Network and Computer Applications 36(2):567–581
31. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Communications in Statistics-theory and Methods 3(1):1–27
32. Cha SH, Srihari SN (2002) On measuring the distance between histograms. Pattern Recognit. 35(6):1355–1370
33. Claise B, Trammell B, Aitken P (2013) Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. RFC 7011 (Internet Standard). http://www.ietf.org/rfc/rfc7011.txt
34. Piskac P, Novotny J (2011) Using of time characteristics in data flow for traffic classification. In: Proceedings of the 5th International Conference on Autonomous Infrastructure, Management and Security, AIMS 2011. Lecture Notes in Computer Science, vol. 6734, pp 173–176. Springer, Berlin
35. Rousseeuw JPJ (1989) A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational Application Math
36. Chen C, Hu J, Meng Q, Zhang Y (2011) Short-time traffic flow prediction with ARIMA-GARCH model. In: 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 607–612). IEEE
37. Ding C, Duan J, Zhang Y, Wu X, Yu G (2017) Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility. IEEE Transactions on Intelligent Transportation Systems 19(4):1054–1064
38. Doherty KAJ, Adams RG, Davey N (2007) Unsupervised learning with normalised data and non-Euclidean norms. Applied Soft Computing 7(1):203–210
39. Shalyga D, Filonov P, Lavrentyev A (2018) Anomaly detection for water treatment system based on neural network with automatic architecture optimization. Retrieved from: https://arxiv.org/abs/1807.07282
40. Ding D, Han QL, Xiang Y, Ge X, Zhang XM (2018) A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing 275:1674–1683
41. Jadidi Z (2015) Flow-based anomaly detection in high-speed networks
42. Lin X, Huang Y (2021) Short-term high-speed traffic flow prediction based on ARIMA-GARCH-m Model. Wirel Pers Commun, pp 1–10
43. Hao W, Yang T, Yang Q (2021) Hybrid statistical-machine learning for real-time anomaly setection in industrial cyber-physical systems. IEEE Transactions on Automation Science and Engineering
44. Ren W, Yardley T, Nahrstedt K (2018) Edmand: edge-based multi-level anomaly detection for scada networks. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) (pp. 1-7). IEEE
45. Khan IA, Pi D, Khan ZU, Hussain Y, Nawaz A (2019) HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems. IEEE Access 7:89507–89521
46. David J, Thomas C (2019) Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic. Computers & Security 82:284–295
47. Khosravi M, Ladani BT (2020) Alerts correlation and causal analysis for APT based cyber attack detection. IEEE Access 162642–162656:8
48. Shi D, Guo Z, Johansson KH, Shi L (2017) Causality countermeasures for anomaly detection in cyber-physical systems. IEEE Trans on Automatic Control 63(2):386–401
49. Haylett G, Jadidi Z, Thanh KN (2021) System-Wide Anomaly Detection of Industrial Control Systems via Deep Learning and Correlation Analysis. In: IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 362–373). Springer, Cham