JOURNAL OF CYBERSECURITY

Journal of Cybersecurity, 2021, 1–10
doi: 10.1093/cybsec/tyab006
Research Paper

Research Paper

A three-tiered intrusion detection system for industrial control systems

Eirini Anthi [1,*] Lowri Williams[1] Pete Burnap[1] and Kevin Jones[2]

[1]Department of Computer Science and Informatics, Queens Building, Cardiff University, 5 The Parade, Roath, Cardiff CF24 3AA, Cardiff, UK and [2]Digital Transformation Office, Airbus, Newport, UK

*Correspondence address. Department of Computer Science and Informatics, Cardiff University, Cardiff, UK. Tel: +44 (0)29 2251 0056; E-mail: anthies@cardiff.ac.uk

Received 3 September 2020; revised 23 November 2020; accepted 21 January 2021

Abstract

This article presents three-tiered intrusion detection systems, which uses a supervised approach to detect cyber-attacks in industrial control systems networks. The proposed approach does not only aim to identify malicious packets on the network but also attempts to identify the general and finer grain attack type occurring on the network. This is key in the industrial control systems environment as the ability to identify exact attack types will lead to an increased response rate to the incident and the defence of the infrastructure. More specifically, the proposed system consists of three stages that aim to classify: (i) whether packets are malicious; (ii) the general attack type of malicious packets (e.g. Denial of Service); and (iii) finer-grained cyber-attacks (e.g. bad cyclic redundancy check, attack). The effectiveness of the proposed intrusion detection systems is evaluated on network data collected from a real industrial gas pipeline system. In addition, an insight is provided as to which features are most relevant in detecting such malicious behaviour. The performance of the system results in an *F*-measure of: (i) 87.4%, (ii) 74.5% and (iii) 41.2%, for each of the layers, respectively. This demonstrates that the proposed architecture can successfully distinguish whether network activity is malicious and detect which general attack was deployed.

Key words: supervised machine learning, industrial control systems, attack detection, intrusion detection system, networks

Introduction

Critical national infrastructure concepts such as manufacturing, smart grids, water treatment plants, gas and oil refineries, and healthcare are heavily dependent on industrial control systems (ICSs). Such systems include supervisory control and data acquisition (SCADA) systems, which are computer systems responsible for gathering and analysing real-time data, distributed control systems which is a specially designed automated control system that consists of geographically distributed control elements, and other smaller control systems such as programmable logic controllers which are industrial solid-state computers that monitor inputs and outputs and make logic-based decisions for automated processes or machines [1]. Historically, ICS networks and their components were protected from cyber-attacks as they ran on proprietary hardware/software and were connected in isolated networks with no external connection to the Internet [2].

However, as the world is becoming more interconnected, there has been a need to connect different ICS networks together and to the Internet, allowing remote access and monitoring functionalities of these systems. As a result, ICS are now subject to a range of security vulnerabilities [2]. According to Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), the number of cyber-attacks against ICS systems has significantly increased over the past few years [3], some of which were of high impact. Such attacks included the Stuxnet attack [4] which targeted the Iranian nuclear enrichment plant and led to physical damages and delayed operations, the Ohio Nuclear Power Plant attack [5] which crashed the safety parameter display system, and the Ukrainian Power grid attack [6] which left approximately 225,000 people without electricity.

Given the importance of these systems, they are an attractive target to attackers. Thus, developing mechanisms that can

automatically detect cyber-attacks in these networks is crucial. Intrusion detection systems (IDS) which monitor and identify malicious behaviour on network traffic have been extensively researched and used in traditional IT infrastructures. However, limited effort has been conducted in the designing and implementation of IDS that are specifically tailored for ICSs [7]. Such tools play a key role in the understanding the cyber-attack that has occurred and can aid a faster and more efficient incident response rate.

ICS networks consist of specific characteristics which make the development of IDSs challenging. First, ICSs have their own protocols (e.g. Modbus, DNP3) which traditional IDSs neglect. Moreover, as these systems are part of critical national infrastructure and handle sensitive processes, accessing the necessary data to test and evaluate a proposed IDS may pose as a challenge. Because of its cyber-physical nature, it is important to have access not only to network/protocol information but also to information related to physical process controls. However, the hardware of these systems is very expensive, limiting the ability to set up ICS testbeds [7].

Applying traditional IDSs to ICS environments would be inefficient as they come with several limitations: (i) most conventional IDSs are signature/rule/event-based which limits the number of attacks they can detect and are inefficient against zero-day attacks; (ii) popular IDSs such as SNORT and Bro are only efficient on traditional IP-only networks and have not been designed to take into consideration ICS-specific protocols [8] and (iii) existing IDSs lack sufficient generality and flexibility to adapt to other systems [7].

To address the aforementioned limitations, this work examines the viability of applying supervised machine learning to detect cyber-attacks in ICSs. A machine learning-based IDS is adaptable and more flexible, as they can automatically learn the general characteristics from data, and thus can form decisions on unseen data [9]. In addition, this approach does not require attack signatures or pre-defined rules to detect attacks, and therefore, it can be effective against zero-day attacks. As a result, this article proposes a three-tiered IDS for the ICS environment which: (i) learns the normal behaviour of the system and identifies malicious activity on ICS/SCADA networks; (ii) identifies the general attack type that has occurred; and (iii) specifies the attack type even further by classifying packets from (ii) as a specific attack type. Being able to detect the generic type of the attack helps security engineers to quickly understand the threat they have to combat. This is because there are many forms of such attacks, that is, Denial of Service (DoS) [e.g. ping flood, ping of death, bad cyclic redundancy check (CRC)]. However, if this detection can be expanded to also identify the exact type of attack which has occurred, it is possible to respond even more efficiently and launch the appropriate countermeasures. To demonstrate the effectiveness of the proposed method, an annotated Gas Pipeline dataset [10], which contains labelled packets from 7 generic attack categories and 35 specific attack categories, was used.

Previous research has mainly attempted to use machine learning algorithms to distinguish between benign and malicious ICS traffic and only one paper has attempted to identify the general attack type that has occurred. Specifically, Beaver *et al.* [11] investigate how supervised machine learning can be used to distinguish malicious behaviour in a gas pipeline ICS. They classify malicious packets as one of seven main attack types. However, further analysis on Beaver's *et al.* [11] gas pipeline dataset showed that there was not enough randomness among normal and attack behaviours [12]. As a result, the machine learning algorithms detected the attacks with very high accuracy (98–100%). This report [13] contains the details that classify this dataset unsuitable for IDS research due to obvious correlations between particular parameters and the result to be predicted. Moreover, the main motivation of this article is not only to detect general attack types but also to distinguish between 35 finer-grained attacks.

According to the Cyber Security Incident Response Guide [14], and National Cyber Security Centre, one of the toughest challenges for organizations is to identify the type of cyber-attack which is occurring on the network without having to perform an in-depth investigation, which can be a very time-consuming process. This is particularly difficult in cases such as ICSs, where the different types of attacks can be very similar (e.g. the slight modification of the pressure values may not be detected) and can have the same initial symptoms. Therefore, in the context of ICS, given that an attack against these systems may have severe consequences and result in hardware damage, injury, environmental impact, or even loss of life, launching specific countermeasures to mitigate these attacks as soon as they occur is critical. As a result, having a mechanism to not only automatically identify malicious packets and their general attack type (e.g. DoS), but also, provides information regarding the exact type of attack (e.g. solenoid attack) is key to a faster, more efficient, and targeted incident response to defend a critical infrastructure. Particularly, the general attack type helps in identifying the implications of the attack. For instance, if a DoS is detected it is consequent that a blackout might be caused. However, knowing the exact attack that is occurring in the system, for instance, a 'Negative Pressure Attack' has been identified, rather than a 'Naive Malicious Response Injection' can significantly assist in locating the attack and defending against it significantly faster by launching countermeasures.

A contributions

Therefore, this article expands on Beaver's *et al.* [11] approach in the following ways:

- The data used to support the experiments provided in this article were presented by Morris *et al.* [10], who document approaches for sharing data for the ICS IDS research community. This dataset was also collected from a gas pipeline ICS but is considered as being more realistic than Beaver *et al.*, as it contains more randomness among benign and malicious scenarios.
- The main contribution of this article is not only to distinguish benign/malicious packets or to identify the general attack type of the malicious packets but to attempt to detect the specific type of the attack that has been deployed by classifying malicious packets as 1 of 35 attack types. As machine learning offers early attack detection, this information would add significant value during incident response by rapidly reducing the time needed to launch-specific countermeasures, and therefore, decreasing the impact of the cyber-attack.
- In comparison to Beaver *et al.* [11], Morris' *et al.* [10] dataset contains more features, and thus in this article, their importance towards identifying malicious behaviour is investigated.
- In this article, 10 supervised machine learning classifiers are evaluated based on previous ICS IDS research [11, 15, 16].

Related work

Several studies concerning ICS security have attempted to investigate how both supervised and unsupervised machine-learning techniques can be used to support the adaptive capabilities of automated IDSs.

In addition to Beaver's *et al.* [11] evaluations, Nader *et al.* [17] use one-class classification techniques which are the Support Vector Data Description and the Kernel Principal Component Analysis for intrusion detection in SCADA systems. They demonstrate that their approach can successfully detect intrusions; however, they do not identify the type of attack which has occurred. Bigham *et al.* [18] investigate how statistical Bayesian networks can be adopted to reduce false positive rates and increase the accuracy of anomaly detection systems in SCADA networks. Moreover, Shengyi *et al.* [19] applied common path mining techniques to develop a hybrid intrusion detection system for power grids. The IDS uses features of signature and specification-based IDSs and is able to classify system behaviour over time, normal control operations, and cyber-attacks. Nevertheless, this work is based on synchrophasor measurement data, which can limit the applicability of this system.

Feng *et al.* [7] developed a multi-level anomaly detection system for ICS, which uses packet signatures and LSTM networks, to successfully detect anomalies in gas pipeline systems. Though do not attempt to classify specific attack types. Parthasarathy and Kundur [20] developed a bloom filter-based IDS for smart grid SCADA, where the regular communication patterns of SCADA and the physical states of power systems have been used to implement lightweight IDS that detects malicious activity. Goh *et al.* [21] proposed a novel unsupervised approach to detect cyber-attacks in cyber-physical systems using recurrent neural networks. They demonstrated that this approach can successfully detect most cyber-attacks with very low false-positive rates. Moreover, Maglaras and Jiang [22] demonstrated that one-class support vector machine (OCSVM) can be promising in detecting anomalies in SCADA communication networks; however, they need to evaluate the proposed system further. Maglaras *et al.* [23] also used OCSVM to implement novel IDS named as K-OCSVM, which has the capability of detecting occurring attacks with high accuracy.

In addition, Pan *et al.* [24] employed a Bayesian network to graphically encode the causal relations among the available information to create patterns with temporal state transitions, which are used as rules in a proposed intrusion detection framework for electric power systems. They demonstrated that the IDS was effective in detecting anomalies on the electric system. Kravich *et al.* [2] used convolutional neural networks to detect cyber-attacks in a secure water treatment plant. They demonstrated that this approach can successfully detect the majority of attacks with low false-positive rates. Linda *et al.* [25] developed an IDS using a combination of neural networks which successfully detected network intrusions in a critical infrastructure testbed. Ghaeini *et al.* [26], employed supervised machine learning algorithms to implement a stateful detector that focuses on identifying stealthy attacks on ICSs.

Furthermore, Gao *et al.* [27] also developed a neural network-based IDS which monitors the physical behaviour of a SCADA system and detects artefacts of command and response injection attacks. Inoue *et al.* [28] compare the efficiency of deep neural networks and OCSVM to detect anomalies in cyber-physical systems. They found that deep neural networks is more efficient and generates lower false-positive rates. Jones *et al.* [29], proposed an SVM-like algorithm which finds a description in a signal temporal logic formula of the known region of behaviours. This approach often creates a readable description of the known behaviours; however, if the system behaviour does not allow for a short description in signal temporal logic, this method will not work.

Finally, there are a few commercially available solutions that employ machine-learning algorithms to detect cyber-attacks provided by companies such as Darktrace [30] and Veracode [31]. However, there is no transparency of the methodology and algorithms employed by these companies, and therefore it is not possible to directly compare this work with these products. Finally, in their documentation, they focus mainly on identifying malicious activity and do not attempt to classify the attack that is occurring on the network.

To summarize these approaches, Table 1 shows existing IDSs for ICS and categorizes them according to detection method, attack type [binary (malicious/benign), general attack (e.g. DoS, reconnaissance), specific attack (e.g. setpoint attack, pump attack), and validation dataset]. We can see that although significant work has been undertaken to identify malicious and benign traffic, only two previous papers have attempted to drill into the attack traffic in more detail to categorize them as general types, and none to date have identified specific attacks. We argue that this information can significantly enhance the incident response process, as knowing the specific attack may lead to launching the most effective and targeted countermeasures.

Regarding the work of the two aforementioned papers that have attempted to classify general attack types, one of them uses a dataset that is not suitable for IDS research, the other one is based on synchrophasor data which limits its ability to generalize to other systems. Finally, although previous research has also attempted to distinguish between benign and malicious traffic, the majority of the methods used are tailored to specific features derived from the specific ICSs (e.g. attributes from train's brake system). As a result, these are not comparable to this work. To the best of our knowledge, this article is the first to use machine learning to not only detect the presence of a cyber-attack but also to detect finer-grained attacks in a Gas Pipeline system.

Methodology

System overview

Figure 1 provides an overview of the proposed IDS architecture. In more detail, on the left, there are various ICS components which generate network data. The data are then being picked up from the IDS tool which constantly listens to the network traffic. The first stage includes the data preprocessing, where the relevant features are being extracted from the network data. At the second stage, the machine-learning algorithm will classify the packets as benign or malicious. If the tool classifies the packet as malicious, then the third and fourth layer will attempt to identify the general attack type and the specific attack type. As a result, in the event of an attack, the output of the proposed system is as follows: (i) benign/malicious; (ii) if malicious the system classifies the packet into one of the seven general attack types it has been trained on and (iii) it will also attempt to identify the specific attack. Knowing both the general attack type and specific attack that is occurring in the ICS environment is critical to better understand the risk and implications of the attack, but also to locate it and defend against it. In order to identify which algorithms are best suited for the implementation of the proposed system, a series of experiments were conducted and discussed in the following sections.

Table 1: summary of current work on IDSs for ICS

References	Detection method	Malicious/ Benign	General attack type	Specific attack type	Dataset
[11]	Supervised	+	+	–	Gas pipeline (2013)
[17]	Unsupervised	+	–	–	Gas pipeline and water storage tank
[18]	Supervised	+	–	–	Electricity management system
[24]	Hybrid	+	+	–	Synchrophasor system
[7]	Unsupervised	+	–	–	Gas pipeline dataset (2013)
[20]	Bloom Filter	+	–	–	ICS Modbus based data
[21]	Unsupervised	+	–	–	Secure water treatment plant (SWat)
[22]	Unsupervised	+	–	–	SCADA dataset
[18]	Specification-based	+	–	–	Power system
[2]	Unsupervised	+	–	–	SWat
[25]	Unsupervised	+	–	–	ICS Modbus based data
[28]	Unsupervised	+	–	–	SWat
[29]	Supervised	–	–	–	Train's brake system
[26]	Supervised	+	– (focused on one attack ZeRA)	–	Secure water treatment plant (SWat)
Current article	Supervised	+	+	+	Gas pipeline dataset (2015)

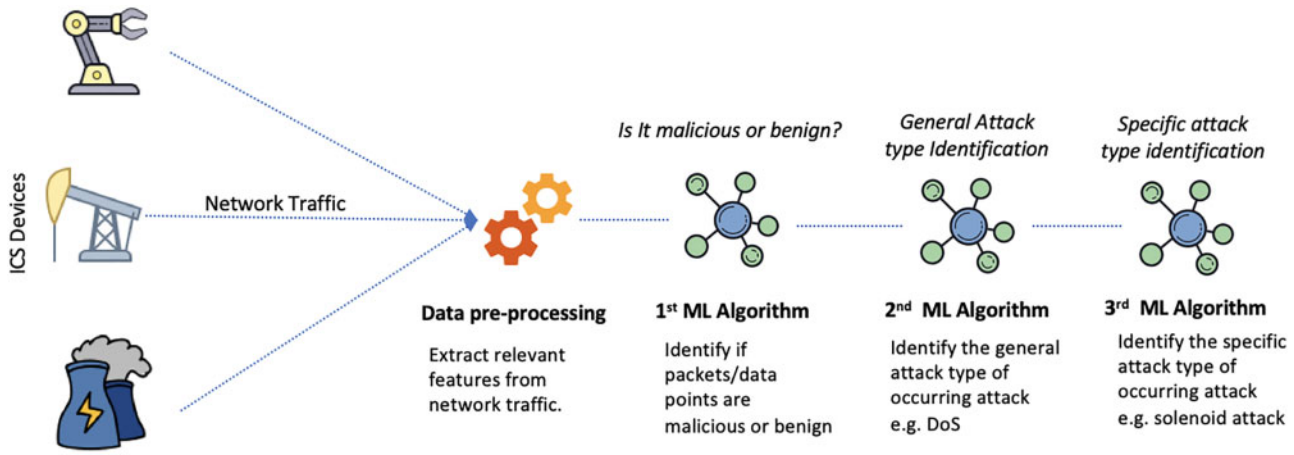

Figure 1: Architecture of the proposed three-tier IDS system for ICS.

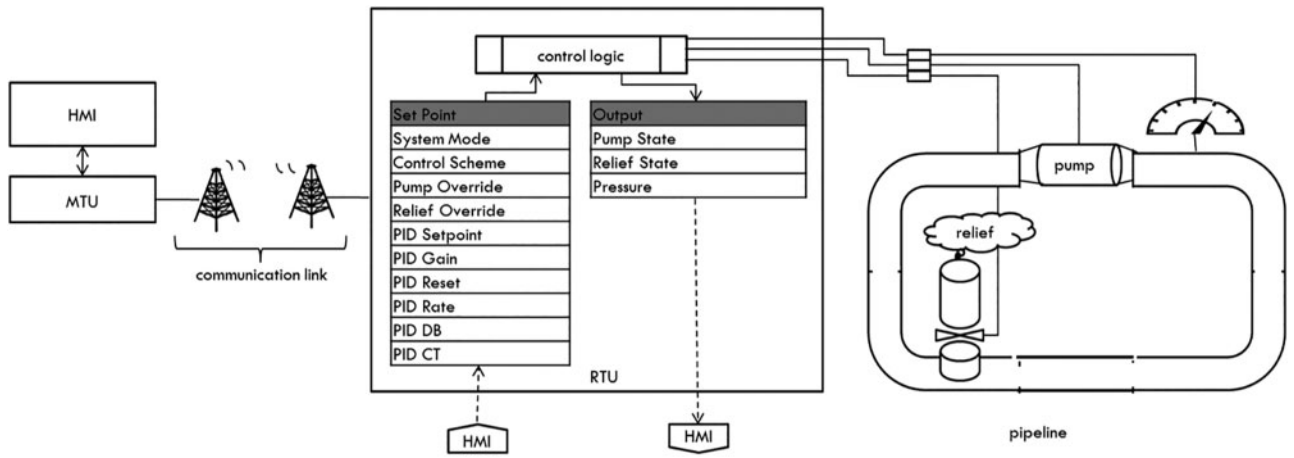

Figure 2: Gas pipeline ICS testbed.

Gas pipeline ICS testbed

Mississippi State University's in-house SCADA lab implemented a scaled-down version of a real gas pipeline system (see Figure 2). The system consists of three major components: sensors/actuators, a communication network and a supervisory control; and operates in three main modes: automatic, manual and off. Its main communication protocol is serial Modbus RTU. This system was used to

generate both benign and malicious data in Turnipseed [12], where more information on the system's specifications can be found.

Data collection

A new framework for collecting data was used to generate the dataset discussed in this work. This new method allowed the creation of a more randomized, realistic and representative dataset. Specifically, to create a more authentic benign dataset, auto IT scripts to simulate real operator activity and to switch between the different operational modes were used. Specific details regarding the generation of the new more realistic dataset are discussed in Morris *et al.* [10]. Similarly, in order to generate the malicious dataset, scripts that randomized and parameterized the launch of a range of attacks were used [12]. The provided dataset represents network packets that were delivered to either the RTU or MTU unit. Each instance in the dataset contains mainly network and payload information.

Cyber-attacks in ICS ecosystems

Multiple studies [12, 32–34] have demonstrated that ICSs are most vulnerable to attacks that fall under four general categories: interception, interruption, modification and fabrication. Specifically:

- *Interception*: Attackers are able to gain information about the devices, their network behaviour, their normal operation, the system information, etc. An example of such an attack is man-in-the-middle.
- *Interruption*: Attackers use such attacks in order to disrupt and, most of the time, make communications between the devices in the ICS network completely unavailable. An example of such an attack is a DoS.
- *Modification*: These attacks allow attackers to alter the values, parameters, or states in a system. For example, in the gas pipeline system, an attacker would have the capability to modify the set-point parameters which control the pressure levels, causing severe damage to the system.
- *Fabrication*: The attacker is able to craft new packets that may seem to be legitimate, but contain altered values that intend on causing damage to the system.

Popular cyber-attacks that fall under the aforementioned categories and thus included within [12] are:

1. Naive Malicious Response Injection;
2. Complex Malicious Response Injection;
3. Malicious State Command Injection;
4. Malicious Parameter Command Injection;
5. Malicious Function Code Injection;
6. DoS and
7. Reconnaissance.

Such attacks may further be broken down into finer-grained attack types. Table 2 describes the 35 specific attacks that were deployed on the ICSs and their effects.

Final dataset

Figures 3–5 show the overall distribution of packets across all classes for each experiment. More specifically, the dataset consists of 60,048 malicious and 214,580 benign packets (Fig. 3). Figure 4 demonstrates the distribution of packets across the seven general attack types, with the (4) 'Malicious Parameter Command Injection' attack having the highest number of packets (20,412) and the (6) 'DoS' attack having the lowest (2,176). Similarly, Fig. 5 demonstrates the distribution of packets across the 35 specific attack types, with (35) 'Slow attacks' having the highest number of packets (2,204) and (20) 'Device scan attack' having the lowest (666).

Supervised machine learning

The experiments presented in this article were performed using Weka [35], a popular and widely used suite of machine learning software.

Feature selection

In order to perform machine learning classification experiments, it is essential to identify which attributes best describe the dataset. In this case, the instances within the dataset contain attributes associated with the RTU's network and payload information. The complete set of features used to evaluate a series of machine-learning classifiers is shown in Table 3. However, for the experiments conducted in this article, features that represented identifying properties were removed (i.e. address and time) to ensure that the model was not making decisions dependent on the specific device or time.

To gain a better insight as to which features are most relevant for distinguishing attack types, a selection filter (InfoGainAttributeEval) was applied. This filter evaluates the worth of an attribute by measuring the information gain with respect to the class. The filter was applied to all attributes for all three different experiments. The results are shown in Table 4.

The results demonstrate that for all three experiments the top three most important features are the 'CRC, the Modbus frame length, and function code values'. Specifically, the CRC allows the system to check for errors within a frame that is sent to either the master or the slave device. An attacker could potentially transmit altered/malicious CRC values to cause attacks such as DoS. The 'Modbus frame' feature is fixed for each command or response query. In the gas pipeline system, a set of write and read commands are used to repeatedly perform block writes and block reads from specific registers. To detect attacks, frames that are not of specific length may be detected as anomalous [12]. Finally, during normal behaviour, the function codes used in the gas pipeline system are usually represented as read (0×03) and write (0×16) commands. However, there exist 256 possible function codes. Some of these function codes can potentially be used for malicious purposes. For example, the 0×08 function code is generally used for diagnostics purposes, but it can be used to force a slave device into a listen only mode.

Conversely, for all three experiments, the bottom three features are 'pump state, solenoid value and control scheme'. Each of these features is represented by binary values. For example, the 'pump state' indicates off (0) or on (1) state. The system can be put into a critical state if an attacker was able to change the system mode to manual and turn the pump on, causing serious physical damage [12]. The 'solenoid' value also has two possible values: closed (0) or opened (1). Similar attacks to the pump may be performed, affecting the system's pressure and causing damage. Finally, the 'control scheme' in the gas pipeline determines whether the system will be controlled by the 'pump' or by the 'solenoid'.

Intuitively, given that the top three features have specific values under normal behaviour, but can accept a range of other values which may indicate abnormal behaviour, such features justifiably influence the classifier in distinguishing whether an attack has occurred. On the other hand, the lowest three features are represented only by binary values which are easier to mask attacks,

Table 2: Thirty-five cyber-attacks in which compromise ICS systems' vulnerability [12]

ID	Name	Type	Description
1–2	Setpoint attacks	MPCI	Changes the pressure set point outside and inside of the range of normal operation
3–4	PID gain attacks	MPCI	Changes the gain outside and inside of the range of normal operation
5–6	PID reset rate attacks	MPCI	Changes the reset rate outside and inside of the range of normal operation
7–8	PID rate attacks	MPCI	Changes the rate outside and inside of the range of normal operation
9–10	PID deadband attacks	MPCI	Changes the dead band outside and inside of the range of normal operation
11–12	PID cycle time attacks	MPCI	Changes the cycle time outside and inside of the range of normal operation
13	Pump attack	MPCI	Randomly changes the state of the pump
14	Solenoid attack	MPCI	Randomly changes the state of the solenoid
15	System mode attack	MPCI	Randomly changes the system mode
16–17	Critical condition attacks	MPCI	Places the system in a critical condition. This condition is not included in normal activity
18	Bad CRC attack	DoS	Sends Modbus packets with incorrect CRC values. This can cause denial of service
19	Clean registers attack	MFCI	Cleans registers in the slave device
20	Device scan attack	Recon	Scan for all possible devices controlled by the master
21	Force listen attack	MFCI	Forces the slave to only listen
22	Restart attack	MFCI	Restart communication on the device
23	Read Id attack	Recon	Read ID of slave device. The data about the device are not recorded, but is performed as if it were being recorded.
24	Function code scan attack	Recon	Scans for possible functions that are being used on the system. The data about the device are not recorded, but is performed as if it were being recorded
25–26	Rise/Fall attacks	CMRI	Sends back pressure readings which create trends on the pressure reading's graph
27–28	Slope attacks	CMRI	Randomly increases/decreases pressure reading by a random slope
29–31	Random value attacks	NMRI	Random pressure measurements are sent to the master
32	Negative pressure attack	NMRI	Sends back a negative pressure reading from the slave
33–34	Fast attacks	CMRI	Sends back a high set point then a low set point which changes 'fast'
35	Slow attacks	CMRI	Sends back a high set point then a low set point which changes 'slow'

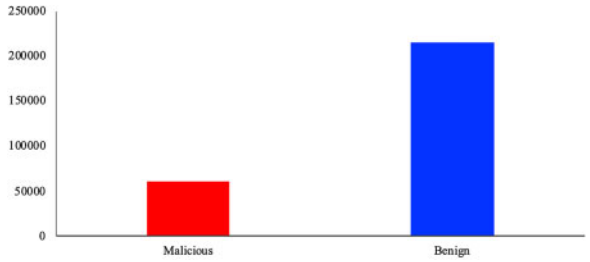

Figure 3: Distribution of packets across attack detection.

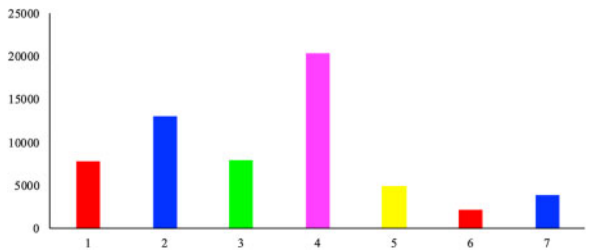

Figure 4: Distribution of packets across seven general attack types.

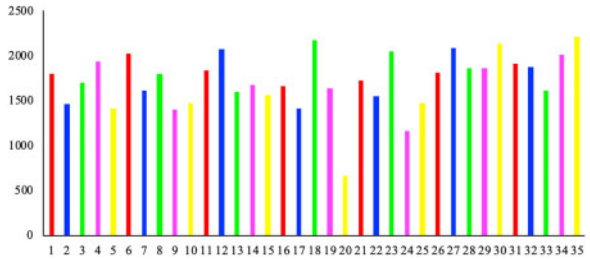

Figure 5: Distribution of packets across 35 specific attack types.

Table 3: Twenty packet features

ID	Feature	Type
1	Address	Network
2	Function	Command payload
3	Length	Network
4	Set point	Command payload
5	Gain	Command payload
6	Reset rate	Command payload
7	Deadband	Command payload
8	Cycle time	Command payload
9	Rate	Command payload
10	System mode	Command payload
11	Control scheme	Command payload
12	Pump	Command payload
13	Solenoid	Command payload
14	Pressure measurement	Command payload
15	CRC rate	Network
16	Command response	Network
17	Time	Network

making it more difficult for the classifier to distinguish malicious behaviour. Understanding which features are most relevant to the classifier is important as it identifies which features must be present in order to best discriminate between the classes. Features which are least relevant to the classification problem may add noise and lead to inaccurate predictions.

Classification experiments

To explore how well classification algorithms can detect cyber-attacks in the ICS environment, the evaluation methodology described in Anthi *et al.* [36] was used.

More specifically, in order to perform classification experiments, a random subset of 60% of each dataset described in 'Use case'

section was selected for training, with the remaining 20% used for testing and 20% used for evaluating the performance of the trained models even further on an unseen dataset. When using the percentage-split function in Weka, the software splits the data so that the distribution of classes in the original dataset is reflected in each dataset produced in the split. In this case, the training datasets for each experiment reflect similar distributions of classes as noted in Figs 3–5.

According to the 'no free lunch' theorem [37], there is no universally best learning algorithm. That is, the choice of algorithm should be based on its performance for that particular problem and the properties of data that characterize the problem. As a result, a variety of classifiers distributed as part of Weka were evaluated.

More specifically, for the specific classification problems considered in this work, 10 classifiers were selected based on their ability to support multi-class classification and high-dimensional feature space. The classifiers included:

- generative models that consider conditional dependencies in the dataset or assume conditional independence (e.g. Bayesian Network, Naive Bayes) and
- discriminative models that aim to maximize information gain or directly maps data to their respective classes without modeling any underlying probability or structure of the data (e.g. J48 Decision Tree, Support Vector Machine).

Moreover, the aforementioned algorithms were also chosen as they produce classifications models that can be easily interpreted, allowing a better understanding of the classification results.

Table 4: Ranked features following info gain ratio attribute filtering

Detecting cyber-attack		General attack type		Specific attack type	
ID	Ranking	ID	Ranking	ID	Ranking
15	0.1532478	2	1.541319	2	1.63269
3	0.0938004	3	1.421765	3	1.46279
2	0.0837514	15	0.97309	15	1.35428
14	0.0379864	16	0.268157	14	0.67725
16	0.0244906	16	0.094191	6	0.38805
6	0.0066772	6	0.0629	4	0.35651
4	0.0052191	4	0.060944	16	0.27546
5	0.0037535	7	0.03744	7	0.25728
9	0.0034525	5	0.013275	5	0.1134
8	0.0030155	10	0.009748	8	0.09489
7	0.0020796	8	0.008695	9	0.08478
10	0.004007	9	0.006817	10	0.03639
12	0.0000764	12	0.004089	12	0.02187
13	0.0000213	13	0.001602	13	0.01008
11	0	11	0.0022	11	0.00345

Results

Tables 5 and 6 report the overall weighted-averaged performance for all 10 classifiers for both the testing and validation datasets, respectively. To gain a better insight into the performance of the classifiers across the experiments, the confusion matrices in Tables 7 and 8, which show how the predicted classes for individual packets compare against the actual ones, were analysed.

Detecting cyber-attacks

When detecting malicious behaviour, the Random Forest achieved the best classification performance with an *F*-measure of 87.4%. Overall, the confusion matrix in Table 7 demonstrates some confusion. This could be explained by the fact that the attacks that were performed during data collection involve altering the values of the core features of the gas pipeline in a discrete manner, for example, changing the 'pump state' from being on or off.

Classifying general attack types

When distinguishing the type of attack among seven attack types, the J48 classifier achieved the best classification performance with an *F*-measure of 74.5%. Overall, the confusion matrix in Table 8 also demonstrates some confusion. In particular, the first ('Naive Malicious Response injection') and the second ('Complex malicious response injection') attacks and the third ('Malicious state command injection') and fourth ('Malicious parameter command injection') are often misclassified. This misclassification can be explained by the fact that such attack types are based upon other attacks, and although they have incurred minor modifications, their compositions are similar.

On the other hand, the fifth ('Malicious function code injection'), sixth '(DoS') and seventh ('Reconnaissance') incur very little confusion. This may be explained by the fact that although normal function codes are usually represented by two values, an attacker can inject up to 256 different values. As a result, this can be easily detected. Finally, reconnaissance activity can also be easily distinguished as it is significantly different from all the other attacks in the dataset.

Table 5: Weighted average classification results across 10 classifiers on a testing dataset

	Detecting cyber-attack			General attack type			Specific attack type		
Classifier	P	R	F	P	R	F	P	R	F
Bayesian network	86.8	87.4	86.8	76.3	74.7	70.1	46.7	37.9	37.5
Naïve Bayes	86.1	83.1	78.1	73.1	58.9	52.8	0.0	23.1	0.0
J48	85.5	86.0	85.7	**78.8**	**76.4**	**73.0**	54.1	42.2	43.2
Zero R	0.0	79.1	0.0	0.0	34.0	0.0	0.0	2.9	0.0
One R	82.3	83.3	80.1	0.0	68.5	0.0	0.0	13.0	0.0
Simple logistic	85.5	82.5	77.1	0.0	71.8	0.0	0.0	26.5	0.0
Support vector machine	NA	NA	NA	0.0	72.1	0.0	0.0	23.7	0.0
Multi-layer perception	83.8	83.7	80.2	64.4	72.3	65.1	0.0	27.3	0.0
Random forest	**87.9**	**88.4**	**87.8**	79.2	75.6	72.4	**58.6**	**43.4**	**44.5**
Decision table	86.0	85.9	83.6	0.0	70.7	0.0	0.0	28.4	0.0

Notes: They highlight the best performing classifiers for each problem.

Table 6: Weighted average classification results across 10 classifiers on an unseen validation dataset

	Detect cyber-attack			General attack type			Specific attack type		
Classifier	P	R	F	P	R	F	P	R	F
Bayesian network	84.4	85.1	84.6	75.9	75.5	72.4	44.8	33.8	34.1
Naïve Bayes	85.4	82.1	77.1	71.9	59.9	54.5	0.0	21.7	0.0
J48	85.7	86.3	85.9	**77.5**	**76.7**	**74.5**	**52.9**	**40.7**	**41.2**
Zero R	0.0	77.5	0.0	0.0	35.0	0.0	0.0	3.2	0.0
One R	79.5	81.0	77.5	0.0	69.5	0.0	0.0	15.1	0.0
Simple logistic	84.3	81.3	75.8	0.0	73.1	0.0	0.0	25.5	0.0
Support vector machine	NA	NA	NA	0.0	73.5	0.0	0.0	23.7	0.0
Multi-layer perception	85.2	83.4	79.7	63.8	72.4	65.3	0.0	25.5	0.0
Random forest	**87.5**	**88.0**	**87.4**	79.1	76.7	74.2	51.8	39.2	39.1
Decision table	84.6	84.7	82.5	0.0	70.7	0.0	0.0	26.7	0.0

Notes: They highlight the best performing classifiers for each problem.

Table 7: Attack detection confusion matrix (random forest)

		Predicted	
		Malicious	Benign
Actual	**Malicious**	11,458	7,100
	Benign	2,785	61,046

Table 8: Identifying general attack type confusion matrix (J48)

		Predicted						
		1	2	3	4	5	6	7
Actual	**1**	1,711	653	0	0	0	0	0
	2	1,263	2,505	0	0	0	0	0
	3	0	0	652	1,805	0	41	0
	4	0	0	176	6,249	0	72	0
	5	0	0	0	0	1,736	0	0
	6	0	0	14	308	0	276	0
	7	0	0	0	0	17	0	1,080

Classifying specific attack types

When distinguishing the specific type of attack among 35 attack types, the J48 classifier achieved the best classification performance with an *F*-measure of 41.2%. Intuitively, this is due to the fact that the classifiers (which are often used for binary classification) are faced with a multi-class classification problem. Thus, further experiments are required to determine whether other approaches, such as ensemble learning, or dividing the dataset according to each attack and evaluating models on each attack type, improve the performance.

The confusion matrix for this classification is too large to be included in this article. However, all attacks from the first to the seventeenth (Table 2) are often misclassified as the eighteenth attack ('Bad CRC Attack'). Decision Tree classifiers operate by splitting the data based on rule/decision boundaries. In the first two experiments, these algorithms seemed to perform very well. However, due to the way it operates, when the algorithm is presented with 35 classes, it creates too many boundaries while not having enough distinct features to base its decisions upon. This might explain why in this experiment its performance is quite poor. Nevertheless, detecting whether a 'Device scan attack, Force listen attack', 'Read Id attack' and a 'Negative pressure attack' has occurred or not demonstrated very little confusion, with all packets being correctly classified.

Use case

Although the architecture of the system proposed in this work has been evaluated on a Gas Pipeline dataset, such an approach can also be applied to other ICSs (e.g. water treatment plants). Intuitively, the features used to evaluate the machine learning classifiers in this article will change depending on the features used to describe the packets collected from other ICS environments.

Moreover, the experiments presented in this article were conducted in an offline setting. This allowed us to investigate the feasibility of the machine-learning approaches. Nevertheless, the positive findings reported herein demonstrate that the proposed system can be implemented as a lightweight machine-learning tool, which can sit on a pipeline to monitor ICS networks and detect attacks in real-time. In more detail, the system can use a network packet sniffer to monitor packets and extract the relevant attributes in order to support the automated classification of malicious packets and their attack types. These results can significantly help in locating the cyber-attack and launching specific countermeasures.

Conclusion

In this article, a novel three-tiered IDS for the ICS environment is presented. The system consists of three stages as follows: (i) identifies malicious packets on the network when an attack is occurring; (ii) classifies the type of the attack that has been deployed from seven main attack types and (iii) specifies the attack type even further by classifying packets from (ii) as 1 of 35 attack types. Currently, only two previous papers have attempted to drill into the attack traffic in more detail to categorize them as general types, and none to date have identified specific attacks. Knowing both the general attack type and specific attack that is occurring in the network is extremely important, as they help understand the risk, impact, and what function has been affected. As a result, they significantly enhance the response and defence time.

To evaluate the performance of the proposed system a range of supervised machine learning classifiers were applied on data from a gas pipeline ICS. The performance of the system's three core functions results in an *F*-measure of: (i) 87.4% (Random Forest); (ii) 74.5% (J48) and (iii) 41.2% (J48). This demonstrates that the proposed architecture can successfully distinguish between malicious

and benign behaviour and detect the general type of attack which has occurred. Although the performance of classifying specific attacks is lower than expected, this initial analysis is promising, as this is the first step towards identifying an appropriate classification approach for specific attacks. This is key in ICS ecosystems, as knowing the exact attack that is occurring can significantly help in locating the cyber-attack and launching even more specific countermeasures.

In addition to classification experiments, the study provides an insight as to which features are most relevant in detecting malicious behaviour and distinguishing among different attack types in ICSs. The findings demonstrate that 'CRC, Modbus frame length, and function code' are the top three most important features which indicate malicious activity in a gas pipeline system. An analysis of the features that are most relevant to the classifier is important as it identifies which features must be present in order to best discriminate between the classes. On the other hand, it least relevant features may add noise and lead to inaccurate predictions. Although the reported results are intuitive, further research and evaluation are required to generalize these findings across other ICS systems.

Funding

This project was part-funded by: Airbus Endeavr, grant "SCADA Cybersecurity Lifecycle 2"; and the Engineering and Physical Sciences Research Council (EPSRC), grant "New Industrial Systems: Chatty Factories", REF EP/R021031/1.

Conflict of interest statement: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Stouffer K, Falco J. *Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security*, 2006.
2. Kravchik M, Shabtai A. Detecting cyber attacks in industrial control systems using convolutional neural networks. In: *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. ACM, 2018, pp. 72–83.
3. N. Cybersecurity and C. I. Centre. *ICS-CERT Year in Review*, 2014. https://us-cert.cisa.gov/sites/default/files/Annual_Reports/Year_in_Review_FY2014_Final.pd.
4. Langner R. Stuxnet: dissecting a cyberwarfare weapon. *IEEE Secur Privacy* 2011;**9**:49–51.
5. Poulsen K. Slammer worm crashed Ohio nuke plant net. *Register* 2003; **20**.
6. Defense Use Case. *Analysis of the Cyber Attack on the Ukrainian Power Grid*. Electricity Information Sharing and Analysis Center (E-ISAC), 2016.
7. Feng C, Li T, and Chana D. Multi-level anomaly detection in industrial control systems via package signatures and lstm networks. In: *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2017, pp. 261–72.
8. Yu T, Sekar V, Seshan S *et al.* Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In: *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*. ACM, 2015, p. 5.
9. Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G *et al.* Anomaly-based network intrusion detection: techniques, systems and challenges. *Comput Secur* 2009;**28**:18–28.
10. Morris TH, Thornton Z, Turnipseed I. Industrial control system simulation and data logging for intrusion detection system research. In: 7th Annual Southeastern Cyber Security Summit. 2015, pp. 3–4.
11. Beaver JM, Borges-Hink RC, Buckner MA. An evaluation of machine learning methods to detect malicious scada communications. In: *2013 12th International Conference on Machine Learning and Applications*, Vol. **2**. IEEE, 2013, pp. 54–59.
12. Turnipseed, IP. A new scada dataset for intrusion detection system research. Ph.D. Dissertation, Mississippi State University, 2015.
13. Turnipseed I. *A New Scada Dataset for Intrusion Detection System Research*. http://sun.library.msstate.edu/ETD-db/theses/available/etd-06292015-115535/unrestricted/final_thesis.pdf (23 November 2020, date last accessed).
14. Csir-procurement-guide.pdf. https://www.crest-approved.org/wp-content/uploads/2014/11/CSIR-Procurement-Guide.pdf (30 July 2019, date last accessed).
15. Tsai C-F, Hsu Y-F, Lin C-Y *et al.* Intrusion detection by machine learning: a review. *Expert Syst Appl* 2009;**36**:11994–12000.
16. Sabhnani M, Serpen G.Application of machine learning algorithms to KDD intrusion detection dataset with in misuse detection context. In Proceedings of the international conference on machine learning: Models, technologies, and applications. . 2003;209–15.
17. Nader P, Honeine P, Beauseroy P. Norms in one-class classification for intrusion detection in scada systems. *IEEE Trans Industr Inform* 2014;**10**: 2308–17.
18. Bigham J, Gamez D, Lu N. Safeguarding scada systems with anomaly detection. In: *International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security*. Springer, 2003, pp. 171–82.
19. Pan S, Morris T, Adhikari U. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Trans Smart Grid* 2015;**6**: 3104–13.
20. Parthasarathy S, Kundur D. Bloom filter based intrusion detection for smart grid scada. In: *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2012, pp. 1–6.
21. Goh J, Adepu S, Tan M, and Lee ZS. Anomaly detection in cyber physical systems using recurrent neural networks. In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2017, pp. 140–5.
22. Maglaras LA, Jiang J. Intrusion detection in scada systems using machine learning techniques. In: *2014 Science and Information Conference*. IEEE, 2014, pp. 626–31.
23. Maglaras L, Janicke H, Jiang J, *et al.* Novel intrusion detection mechanism with low overhead for scada systems. In: *Security Solutions and Applied Cryptography in Smart Grid Communications*. IGI Global, 2017, pp. 160–78.
24. Pan S, Morris TH, Adhikari U. A specification-based intrusion detection framework for cyber-physical environment in electric power system. *Int J Netw Secur* 2015;**17**:174–188.
25. Linda O, Vollmer T, and Manic M. Neural network based intrusion detection system for critical infrastructures. In: *2009 International Joint Conference on Neural Networks*. IEEE, 2009, pp. 1827–34.
26. Ghaeini HR, Tippenhauer NO, and Zhou J. Zero residual attacks on industrial control systems and stateful countermeasures. In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019. pp. 1–10.
27. Gao W, Morris T, Reaves B, and Richey D. On scada control system command and response injection and intrusion detection. In: *2010 eCrime Researchers Summit*. IEEE, 2010, pp. 1–9.
28. Inoue J, Yamagata Y, Chen Y *et al.* Anomaly detection for a water treatment system using unsupervised machine learning. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 1058–65.
29. Jones A, Kong Z, Belta C. Anomaly detection in cyber-physical systems: a formal methods approach. In: *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 848–53.
30. Darktrace: World-Leading AI for Cyber Security, https://www.darktrace.com/en/ (29 January 2020, date last accessed).
31. Veracode: Application Security Software. Use Veracode to Secure the Applications you Build, Buy, & Manage. https://www.veracode.com/ (29 January 2020, date last accessed).

32. Drias Z, Serhrouchni A, and Vogel O. Taxonomy of attacks on industrial control protocols. In: *2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*. IEEE, 2015, pp. 1–6.
33. Stouffer K, Falco J, Scarfone K. Guide to industrial control systems (ICS) security. *NIST Special Publication* 2011;**800**:16–16.
34. Maynard P, McLaughlin K, Haberler B. Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks. In: *2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014) 2. 2014*.
35. Weka 3. *Data Mining with Open Source Machine Learning Software in Java*. https://www.cs.waikato.ac.nz/ml/weka/ (06 March 2018, date last accessed).
36. Anthi E, Williams L, Slowinska M *et al.* A supervised intrusion detection system for smart home iot devices. *IEEE Internet Things J* 2019;**6**: 9042–53.
37. D. H W. The supervised learning no-free-lunch theorems. In: *Soft Computing and Industry*. Springer, 2002, 25–42.